Why Do Students in Computer Science Courses Cheat?

Farah Alyasari
Barrett, the Honors College
Spring 2019
Tempe, Arizona
farah.alyasari@yahoo.com

Abstract : The goal of this study is to equip administrators and instructors with a deeper understanding of the apparent cheating problem in Computer Science courses, with proposed solutions to lower academic dishonesty from the students' perspective.

Keywords : Academic Integrity, Computing Education

Table of Contents

I. BACKGROUND

Academic dishonesty is prevalent in Computer Science courses across the country. A report by Bidgood and Merrill disclosed that Brown University recorded 49 cheating incidents in 2016, and over half of those were in computer science; over 60 students at Harvard University were accused of cheating in its well-known Computer Science 50 (CS50) course in 2016; and five students were accused of cheating in computer science around the same time at Yale University [1]. Although Computer Science students constitute 6.5 % of the student body at Stanford, "of 123 honor code violations investigated [in 2009] by Stanford's Judicial Panel, 28 involved Computer Science students" [2]. And, at least 20% of students in a Computer Science course at Stanford University were accused of cheating in 2015 [1]. To give more examples of how relevant cheating occurs in Computer Science courses, consider the recent two incidents at Columbia University and Penn Engineering College. In the fall of 2016 and spring of 2017, the percentage of students caught cheating in Columbia Advanced Programming Class was 20% and 8% respectively [3]. At Penn Engineering, "45 % of reported cases of academic dishonesty came from the introductory level classes of the CIS department", particularly from CIS 110 Introduction to Computer Program [4]. Even though these data show a high prevalence of cheating in Computer Science Courses, they could also be an underestimation, as they are shared from the perspective of college faculties alone.

II. INTRODUCTION

Restating [5] thoughts on cheating, “Nobody goes into teaching to be a cop”, but whether instructors like it or not, “enforcing standards of academic integrity is a central responsibility of teachers”. To enforce academic integrity in Computer Science courses, one must first understand the underlying motives which cause students to cheat. A strong understanding will help set the foundation for effective solutions to increase academic honesty. The exact purpose of this paper is to give a modular answer as to why students in Computer Science courses cheat. A modular answer is a framework that can be used to understand the variety of reasons which motivate students to cheat. Essentially, it can be used to understand, detect, and make solutions to various cheating incidents. After presenting a modular framework to why students cheat, proposals will be made on ways to guide students into academic honesty.

The remainder of this paper is composed of five sections. The Methods section discusses the design of the questions in the survey. Results from the survey are presented in groups, stated the same way as in the survey, and all the raw data points are shared for both numerical and textual results. The analysis section will guide the reader through the formation of a framework, which gives a modular answer to why students cheat, finding pressure, opportunity, and justification as the three components. This framework is used in business settings to understand

fraud at the workplace, and this research demonstrates that it can also be used as a valuable tool when addressing cheating at the university level. After the discussion, the proposed solution section will highlight strategies that administrators and instructors can implement to increase academic honesty in computer science courses. Comments received from students during this research influence the proposed solutions. The conclusion will summarize key findings from this research, and the significance they have on how instructors and administrators address cheating. To make the most of reading this paper, the reader is encouraged to read the sections in the same order in which they are presented.

Throughout this paper, the term 'Academic Integrity Policy' is used. To be clear, the Academic Integrity Policy (AIP) discussed in this study is the one used by Arizona State University (ASU), which is also adopted by other universities. The AIP divides academic dishonesty into five broad areas: cheating on an academic evaluation or assignment, plagiarizing, academic deceit, aiding academic integrity policy violations and inappropriately collaborating, and falsifying academic records [6]. With respect to the five areas of the AIP, this study will specifically explore collaboration on programming assignments, reasons students report or do not report cheating incidents, and the usage

of external resources to complete programming assignments. The AIP prohibits excessive collaboration and limits resources that students can use to complete their assignments, and this study examines the attitude of computer science students regarding these prohibited activities. Serving a similar purpose as the AIP, the Student Honor Code is an expectation placed on students to uphold the Academic Integrity Policy (AIP). For the purpose of this paper, acts considered as cheating are those incidents in which students violate the Student Honor Code by not following the Academic Integrity Policy.

III. METHODS

In this study, a survey was conducted to help understand why students in Computer Science courses cheat. The survey was available to students in all computer science courses at Arizona State University; however, it was primarily distributed to students in the following courses:

1. CSE 110 – Introduction to Programming

2. CSE 205 – Object-Oriented Programming and Data Structure

3. CSE 420 – Computer Architecture

4. CSE 423 – Capstone for Computer Architecture

All courses above have programming assignments, and the distribution of the survey in both lower and upper division courses ensures that responses reflect a consensus of both incoming Computer Science students and those who have been in the Computer Science program for several years. .

The survey had 27 questions total. Five questions collected data about students' major, course level, professional experience, and opinion about the computer ethics course offered at the university. The remaining 22 questions were divided into four categories. Four questions were about the Academic Integrity Policy, Six questions were about students' lifestyle, Six questions were about students' evaluation of instructors, and Six questions were about students' evaluation of themselves and each other.

In order to collect reliable data in an efficient way, questionnaires in this research survey consist of rating scales, dichotomous, and open-ended questions. A series of questions in the survey utilized an even Likert scale, meaning the neutral option was not available. Removing the middle option from a Likert scale eliminates possible misinterpretation of the midpoint and forces respondents to be more thoughtful [8]. Another set of questions had an odd Likert-scale, as those questions inquired about frequencies of certain student experiences. To lower both incorrect selection and incident of non-response bias, completing all the survey questions was not required to submit the survey, [9]. Open-ended questions were used, as they offer a more respondent-focused survey and more understanding of the data [10]. Comments collected from respondents were searched for possible solutions to address cheating and motivation to report or not report cheating incidents.

In addition, numerical data was used to help understand why students cheat. The Percentage values of Likert scale questions have the following even response items: "Strongly Agree", "Agree", "Disagree", and "Strongly Disagree". The values for "Strongly Agree" and "Agree" are grouped together as general agreement; similarly, "Strongly Disagree" and "Disagree" are grouped together as general disagreement. These questions were for opinion-based

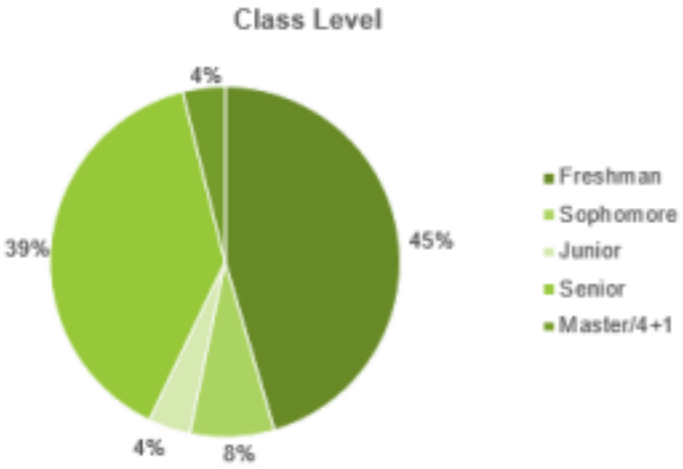
responses. Questions with odd response items have the scale: “Almost Never,” “Seldom,” “Sometimes," “Often,” and “Almost Always.” The “Seldom” and “Almost Never” may be grouped together, and “Often” and “Almost Always” may be grouped together in the analysis. If they are grouped together, then it will be pointed out. Odd likert scale questions were used to measure frequency of certain behaviors. Because of the limited response size, this organization allows for a more meaningful grouping of the data. However, data in the Results section shows the raw values without grouping of the data.

The numerical and textual data discovered from this survey have been viewed in both positivist and interpretivist lights. The interpretivist approach is conducted using open-ended, optional, and unstructured survey questions which encourage computing students to share their own experiences and thoughts regarding the apparent cheating problem at their institution. On the other hand, the positivist approach utilizes numerical data to make conclusions about students’ habits. Data was analyzed for correlations among students of similar grade point averages and course levels, and percentage values were used to find consensus among students. The combination of positivist and interpretivist analytical approaches contributes to findings that would otherwise be missed by adopting only one research approach [7]. The next section will share the survey results, supporting the idea that both numerical and textual data helped in building comprehensive understanding of the underlying cheating problem.

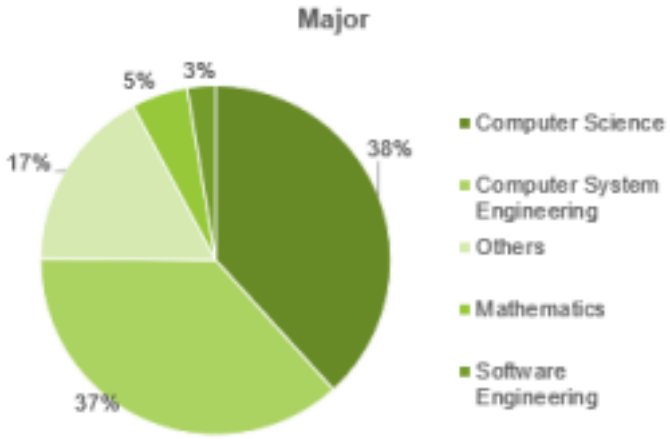
V. RESULTS

A total of 77 responses were received from students in both upper division and lower division Computer Science courses. Figure [1] and [2] show a breakdown of respondents based on course level and major,

respectively. Most respondents come from the freshmen class and senior class. Because of low responses from sophomores and juniors, the sophomores are grouped with the freshmen, and the juniors are grouped with seniors, as groups of lower division and upper division students. Responses from 4+1 and Master students are omitted from comparison between upper division and lower division students because they do not belong to either category. However, they are included when discussing the overall group of respondents because these students were taking the same programming courses as Computer Science undergraduate students, and their insight is similarly valuable. In figure [2], students in the "Others" category belong to majors which require programming courses but are not Computer Science or Computer Engineering majors. Examples of such majors that require programming courses are: Biomedical Engineering, Biology, Industrial Engineering, and Engineering Management.



Figure[1]



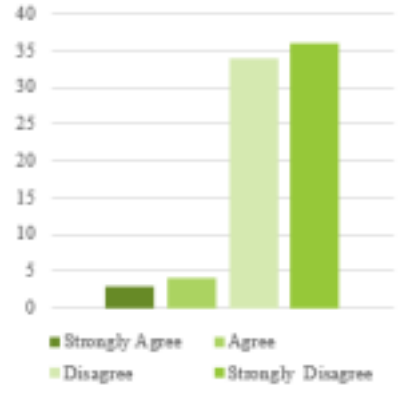
Figure[2]

Survey questions assessing for similar issues were scattered in different pages when distributes to students. However, the results here show the questions by the corresponding categories. Recall, questions were divided into four categories. Five questions collected data about students' major, course level, professional experience, and opinion about the computer ethics course offered at the university. Four questions were about the Academic Integrity Policy, six questions were about students' lifestyle, six questions were about students' evaluation of instructors, and six questions were about students' evaluation of themselves and each other.

All survey questions received between 76 to 77 responses; except the question in figure 9. That question received 50 responses, as it was posted while the survey was live, and after 27 submissions where received. It is worth noting that dichotomous, even Likert, and odd Likert scale questions are scattered in the across different group of questions. A. Students Evaluation of Instructors

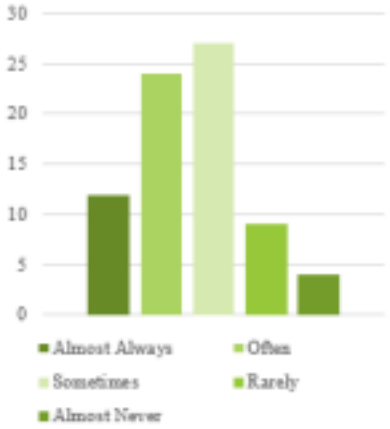
Although instructors ask for course evaluation every semester, questions in this survey can offer more insight, as those responses are gathered from a student-led survey rather than a faculty survey.

Figure[3]



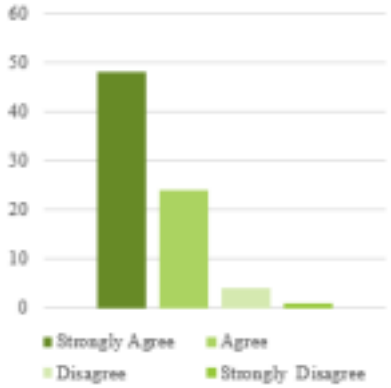
If I am graded unfairly, then It is okay for me to complete my homework unfairly

Figure[4]



My instructors provide learning materials that are easy to understand for someone who is still learning the subject

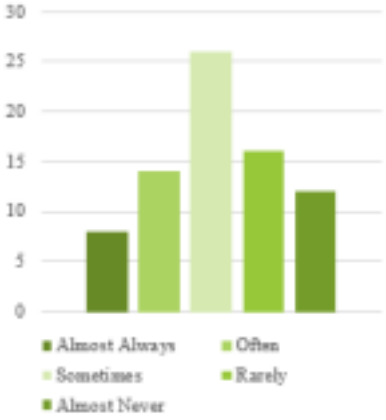
Figure[5]



I would like to see a solution to a programming assignment after the assignment is due

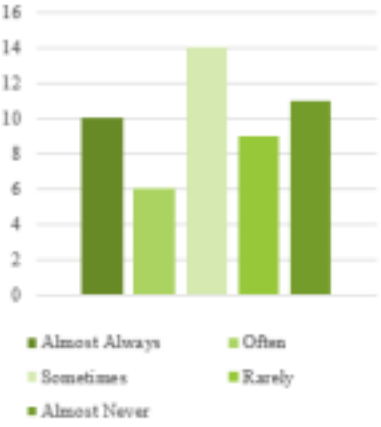
8

Figure[6]



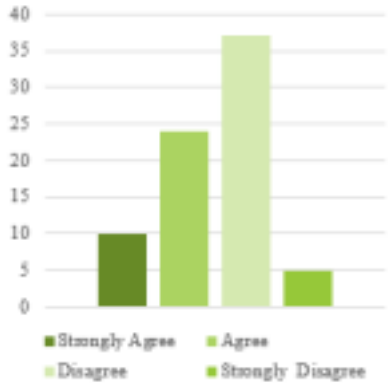
I receive helpful feedback on my programming skills

Figure[7]



My teachers provide a solution to the programming assignment after it is due*

Figure[8]

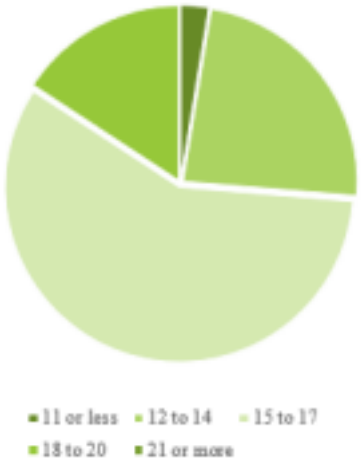


It is not fair for my programming assignment to be graded only on test cases

B. Students' Lifestyle

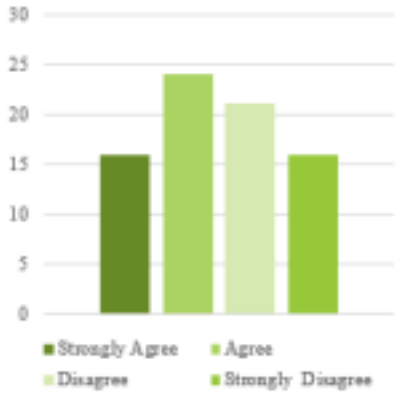
The following questions put numbers to the lives of students, which helps instructors have a better idea of an average student's schedule. This is especially important because in the survey comments, some students explained that instructors make assignments without consideration to students' schedules.

Figure[9]



How many credit hours are you taking this semester?

Figure[10]



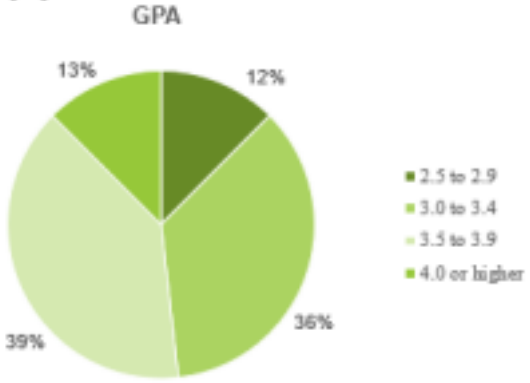
My parents or family members pressure me to get higher grades

Figure[11]

I work on regular basis to provide financial support to myself and/or family

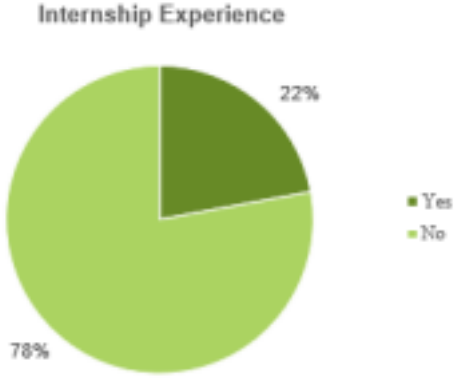
9

Figure[12]



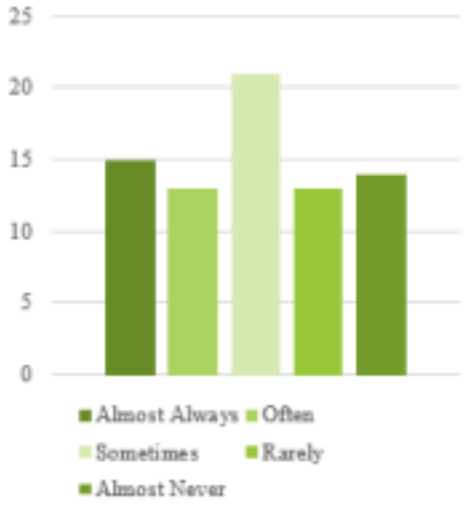
Please select the range of your most recent GPA

Figure[13]



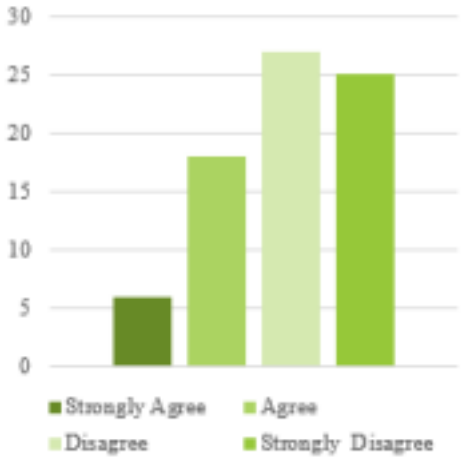
Have you had an internship before?

Figure[14]



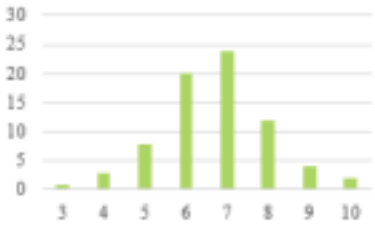
Concerns about my grades impact my ability to sleep, eat, or socialize with my family and peers

Figure[15]



I have personal circumstances which prevent me from studying to my best ability

Please estimate the number of hours you spend on each of the following activities:



Figure[16] Sleeping

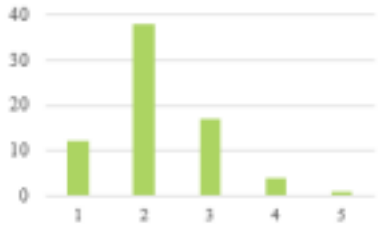


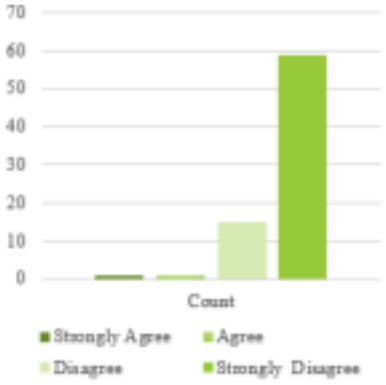
Figure [17] Eating

Figure [18] Studying + HW

C. Students on the Academic Integrity Policy

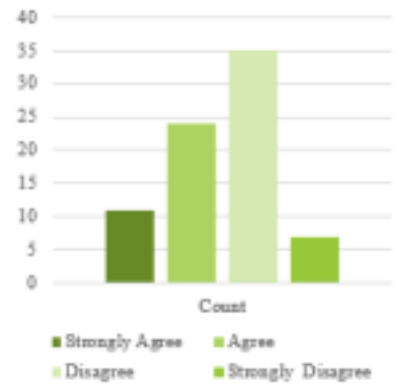
The Academic Integrity Policy is used by the university to define and punish cheating. It is crucial to understand how students view this policy as it can affect whether they truly abide by it. In addition to scale questions, this section has one poll. In this poll, participants were presented with a scenario in which another student read leaked programming assignments on GitHub. Then, participants were asked to select all options which represent their opinion. Data from this poll reflects both students views on the Academic Integrity Policy and their internalized definition of cheating.

Figure[19]



I must cheat in order to pass

Figure[20]

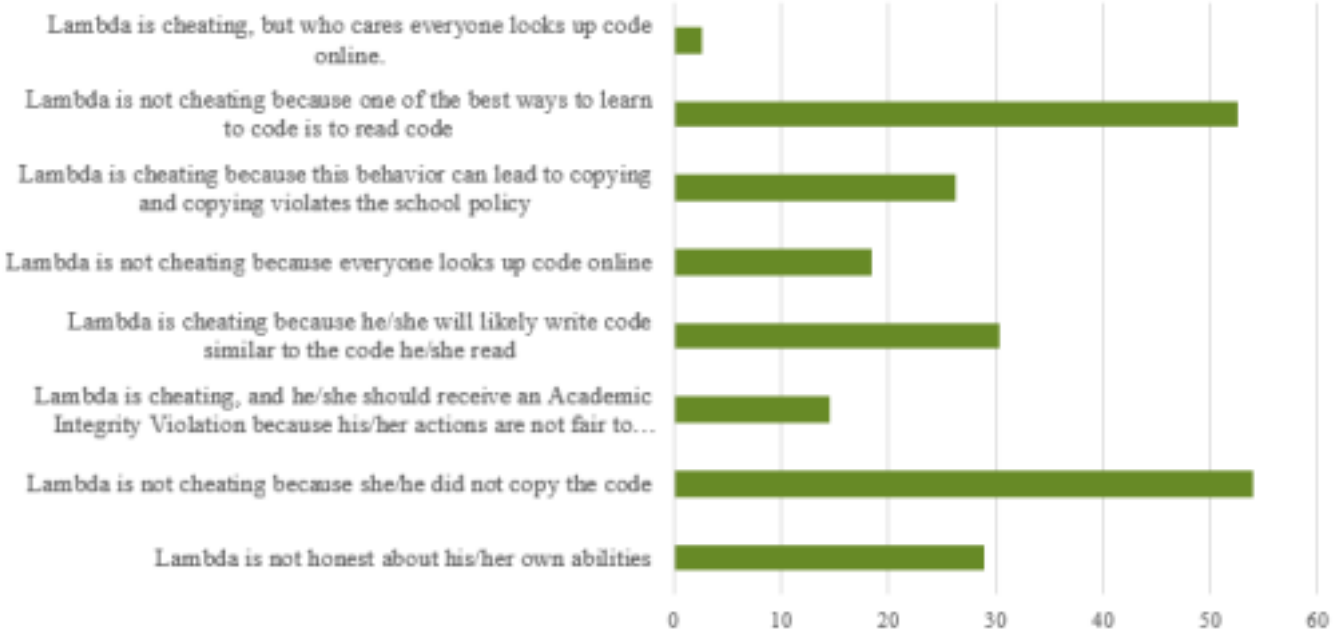


It is fair for me to use any and all resources to complete my programming assignment, with disregard to any and all school policies, as long as i actually learn the materials at the end

Figure[21]

Taking the Computer Ethics course helped me evaluate ethical conflicts in the Computer Science and Computer Engineering Discipline

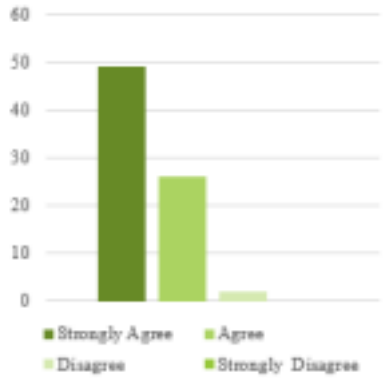
Figure[22]



D. Students Evaluation of Themselves

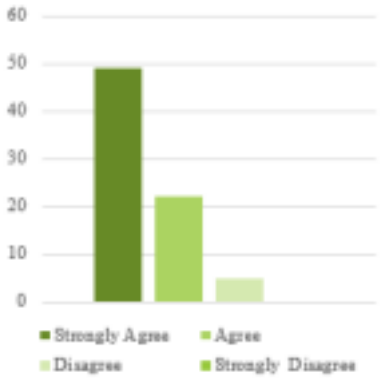
The remaining scale questions ask students to evaluate their skills, and how they feel toward other students, which allows instructors to understand students' culture.

Figure[23]



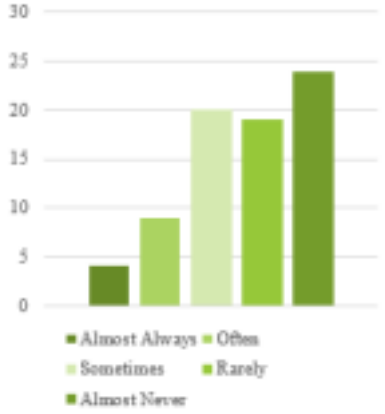
I can correctly explain my code from a randomly chosen programming assignment that I submitted

Figure[24]



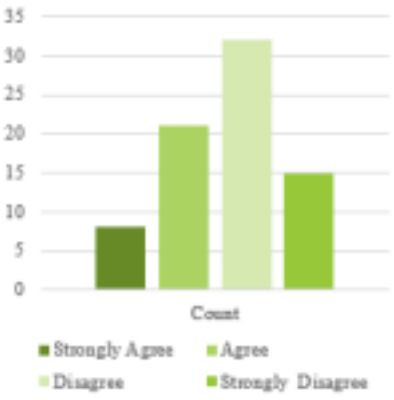
Learning the materials is just as important or more important than earning high grades

Figure[25]



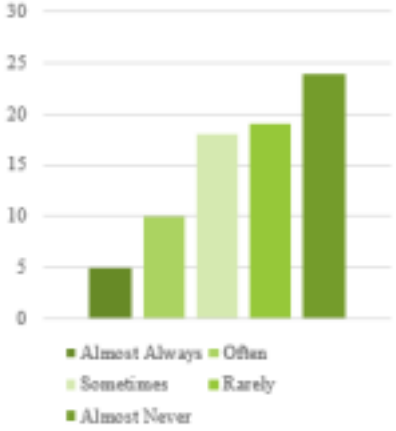
I struggle with programming techniques in my current classes, because I did not practice them enough in earlier classes

Figure[26]



I feel jealous of people who score higher grades than me

Figure[27]



I collaborate with a group of other students on individual programming assignments

Figure[28]

If you catch one of your classmate's cheating, would you report him/her?

E. Students Comments

Perhaps, the more interesting responses come from comments that students shared on the survey. There was a total of 92 comments for two open-ended questions. Some comments were lengthy; however, they are a core aspect of this

paper. After all, this paper intends to shed light on the cheating problem from the students' perspective. This section is a list of Comment for the two open ended questions, and they are listed in the same order in which they were received. For question 1, please see Figure [28] for numerical summery, and this section has the list of comments. All comments are presented as they were; even though, some have spelling errors. Referring to the previous section of students' evaluation of themselves, comments from students provide a deeper understanding of the general attitude and personalities of students taking Computer Science courses.

1) If you catch one of your classmate's cheating, would you report him/her? Why or Why not?

•No his lack of skills will become apart in later courses or in industry
•No I will not
•No, cheating is okay.
•I would not. I dislike being involved in things that don't affect me, so as long as the code that they copied wasn't mine, in my opinion its none of my business
•Usually not because I would ruin their life and future
•No, it's not my place
•Yes, if I had evidence to prove it and it was blatant cheating. Clear cheating affects the whole student body. •Depends on how egregious their cheating is. If it is a case like the Lambda example I probably won't care. •No
•Yes, I wouldn't put up with that behavior.
•No, although I think cheating is wrong , I believe it is equally unethical to interfere in someone else's business. If they are cheating they will be caught.
•No, none of my concern. Their grade is between them and the instructor
•Depends on the kind of "cheating". If they read code to understand it and then write their own I would let it pass but if it is just plain copying then I would probably talk to them first and if they don't change, report them.
•Academic Dishonesty is on its own a self-punishment, as knowledge is not acquired. Denying oneself knowledge is punishment in itself
•No, none of my business.
•Usually, though mostly if it involves consulting other students
•Most likely not, I do not care about other people's grades. Other people's grades matter to me only if the professors curve the grades, but I still don't care enough to report them.
•Probably not, wouldn't affect me
•No, I'm more focused on my own performance than someone else's.
•Probably not, honestly, because it is none of my business how others choose to learn. Also I never assume someone else's circumstances, as they might be referencing something they've already encountered and are refreshing themselves. For a specific assignment, going and looking for the solution is cheating, but if you've already come across a particular piece of code, I think that calls for consideration. No one has every code they've ever seen written/worked on memorized
•No. The school is so chalk-full of Chegg-ers that reporting every instance of the practice would prevent me from doing my own work.

•I would if it directly involved me, because I want to keep my record clean. Otherwise, I would ask them why and try to help them complete their assignments another way.
•No, I don’t the circumstances he/she is undergoing.
•I personally think the only time i would report a student is if they copied from me because that would impact me
•Not my business, I have seen plenty of students and classmates cheating. I actually feel at a disadvantage for not cheating.
•I would report them because they did not put any work into the assignment but might get a good grade which is unfair.
•If there is a way to do it anonymously I would because it's not fair for other students who are honestly completing assignments/tests in the course to receive the same grade if they're cheating. •NO! because no one likes a snitch
•yes
•If seriously cheating yes
•No. I do not want to be the reason behind their bad days
•No, karma
•No, I have too much stress focusing on my own affairs anyway, and that would just add to it •I've never been one to report something like that. They're a fraudulent person and if they come up to me and talk about their interview with Google I'm going to be upset, but I've never been in a situation where I've thought to report it. I'd rather just talk to a therapist.
•Typically not, depends to what extent the cheating is ie on a test I would vs looking up how to do an assignment I probably would not. I understand that not all situations are as they appear, and if it's something they do regularly then when they graduate they aren't going to be able to use their degree because they won't have learned anything. I think that becomes it's own punishment.
•No, cause I think only teacher can tell whether he/she is cheating or not
•Yes because it wrong and in a way also affects my grade
•If the cheating is explicit and nonchalant, I absolutely would.
•No, students' honor code.
•I would first confront them about it before reporting them. If they still cheat I would report them. •Maybe
•Yes, it’s not fair to students
•Yes, because it devalues all the degrees at asu
•Yes
•Yes, cheating is wrong.
•Would let them know it's wrong.
•I aint no snitch
•I would report them because it's unfair that the classmate would get a higher advantage over students who actually tried and it may create a cheating habit in the classmate.
•I wouldn't. It isn't my business.
•Depends on personal charisma。
•maybe not

•I would not because it is not easy and any help is good help
•Probably not, because it doesn't affect me personally
•No, because snitching is very frowned upon where I am from, but I will not assist in cheating in any way,

shape, or form and will remind the classmate of the consequences

•Depends on the severity, reasoning, and effect on me

•Yes. My major depends on ASU. If someone cheats and gets away with it, then my major means less then it did before. I am also working hard to earn my grades. If you're cheating, then why even bother studying to become an engineer?

•Depends

•If it was on a major exam or final, yes.

•No. I don't want to interfere in others work/life.The cheater will get caught at the end of the day what so ever.

•No, I mind my own business

•Yes, I would. Because he is not showing any efforts as other might.

•No because I could not provide proof

•Depends on the situation

2) Is there anything that you would like to share or comment about regarding the cheating problem?

•This entire survey felt so biased why do you hate people who know how to cheat

•I think sometimes the assignments that professors assign are developed in a vacuum that assumes their class is the only class their students are taking. Even if the assignments were only broken up into much smaller sections that build on top of each other I think that a lot of people wouldn't have to worry as much •Everyone is lazy and cheats in the 'small ways'

•They are hurting themselves more than anything

•Apart from the students that just copy others code, I believe that people start "cheating", reading code, because the assignments given are too complicated for the amount of time given to complete them. Sometimes the amount of time given does not account for other classes the student might be taken. •Getting a zero for code that doesn't compile is never an accurate assessment of knowledge. Content is more important

•Those that cheat eventually change their major

•I think it’s way more prevelant than people are willing to admit. I’ve been asked for entire semester work loads (files) for several classes now. I’ve never shared my own work but I have walked people through the steps on how to complete it. This was the case a few weeks ago for 325. Someone asked me if they could see my code and I just reworked the problem out on paper and showed them my method of conpleting the problem without code

•Incompetence is it's own punishment.

•I think the best way to check is to ask the students to explain their code, but i think that would add too much workload for the amount of code and how lengthy they can get.

•Some professors actively encourage students to use Stack Overflow as a resource, the ethics of "looking up code" depends entirely on the context of the assignment (using other libraries for an open-ended 400 level assignment vs referencing a linked list library for an assignment about implementing linked lists in

programming 101)

•No, I feel that your survey covered all aspects of this issue.

•Be more lenient when people are helping each other and give me examples of correct code so I don't have to think of some myself to show other people for help.

•All the best for your thesis

•I do not cheat on my programming assignments and do not consider modifying similar code or reading other people's completed code to be cheating. If there was more emphasis on being able to create code slowly and test it until it worked fully (incremental design) rather than having strict deadlines (that often fall on the same due date as other assignments in other classes) and unapproachable instructors.

•It's everywhere. People get jobs and internships through it. Then they become competent programmers at their jobs. It's like benzodiazepenes, antidepressants, and adderall. I think the person has issues that they can hide at the interview, they're a fraudulent person at their core and have built their resume off of that, they can lie face-to-face to someone for a few minutes when they're asked about 'struggle' they've gone through or whatever. People can mature at a work place. They've shown they're willing to cheat to succeed so maybe it's just a sign that they're willing to 'do what it takes' to get something done. I think/hope what's more likely is that they have an existential crisis in the middle of a job when the pills stop working and they lose motivation to keep succeeding and it all comes crashing down.

•I think cheating in Computing majors is a tricky subject to handle due to the nature of our degrees. As computing majors, typically we know how to find things online and grow from what we do, but it's easy to just find answers and truly be the equivalent of cheating. So it becomes distorted of where should we draw the line? Does it make sense to stop a degree that is so heavily influenced by information online, and then restrict it to book work? Should it change at a certain level? (300+ you've learned the basics and it isn't cheating anymore?) Best of luck!!

•I think that getting help from fellow students should be allowed, as long as they aren’t copying one another’s work.

•No

•I didnt even know there was one. I just feel like too many people choose CS as a major without knowing was it entails

•A lack of supplemental materials and a lack of a way to quickly ask questions will lead to more students cheating.

•If the assignment is tough then people can help you to succeed if not then you are wasting an important resourse

•The only cheating I've heard about is two people helping each other on individual assignments

•Honestly there is no reason to cheat on any of the assignments. If you do what you are supposed to do its

fairly easy

•I feel like there is a lot of overreacting on the faculty's part which leads to underreacting on the students part •It is much easier to cheat when the classes are easy. When you hit 300 level classes, that's where it gets difficult to cheat (I think). At this point, it's inevitable it happens, but they'll get rooted out at some point. Generally.

•No.

•If there were more office hours, I believe it would encourage students to use them, rather than using a coding website with answers.

•N/A

Reading these comments must have been time consuming, but please do not stop reading the paper yet. The next section will analyze results in different categories including students' stance on the Academic Integrity Policy, attitude toward each other and instructors, and students' quality of lifestyle. The analysis will guide the reader through the formation of a framework, which gives a modular answer to why students cheat. The components of the framework will be incrementally revealed after each corresponding analysis. A modular answer can be used to understand the variety of reasons which motivate students to cheat. Not only can it be used to understand why students cheat, it can also help detect, and make solutions to various cheating incidents by abstracting the details that motivate different students to cheat.

A. Students' Evaluation of Instructors

Computer Science students understand that grading a computer program can be tedious. Most students, 55% , believe it is to grade a programming assignment only on test cases; 44% believe that it is not fair. Grading based on test cases entails giving the program several inputs, with expected outputs, and checking if the expected outputs match the real output. Although students are divided about the fairness associated with scripted grading using test cases, they collectively agree that instructors should share programming solutions. 93.5% of students say that they would like to see a solution to a programming assignment after the assignment is due. However, despite the high demand for programming solutions, only 32% of students say that their instructors almost always or often provide a solution to the programming assignment. 40 % of students say their instructors almost never or rarely release a solution, and 28 % say instructors give sometimes solutions. Simply put, students are not certain if they will ever see at least one correct way to solve a problem that they do not know how to solve, or another way to solve a problem that they already know how to solve, unless they search for the solution online. As such, it would become unnecessary for students to search for online solutions, if instructors gave the solution in class.

From the professor's perspective, the 40 % of students who say that their instructors almost never or rarely share a solution might be an example of students who do not seek help. During office hours, instructors might be willing to walk students through a homework question, discuss different approaches to solve a problem, or even show the solution to a specific problem, if the student asks for it. There are also other ways students can seek help; the teaching assistants and tutoring centers are available to help students in-person or online. After all, it is the student's responsibility to try and initiate communication with their instructors to inquire for help. The more students ask for help from their instructors and learning assistance, the less necessary it will be to seek help elsewhere, from sources that are not in line with school policies.

However, it is also difficult for instructors to provide extra individual assessment to every student who wants to see the solution during office hours; it is better if the instructor simply shares the solution with the entire class. Looking back at the results, the 40 % of students who say their instructors almost never or rarely release a solution plus the 28 % who say solutions are given sometimes make up about 68 % of students. It is ineffective to wait in line at office

hours to ask permission to view the solution, when students can find a similar programming assignment online. Some instructors fear that students will leak the solution online and make the problem worse. However, if instructors are willing to show the solution to the entire class, then students will gain more knowledge and spend less time searching for complementary resources online. If instructors insist that they will only show the solution individually, rather than to the entire classroom, they should understand that it will be unfeasible to accommodate all or even most students. It is no surprise that students turn to online resources to complement their education, rather than seek help from instructors.

In addition to inadequate solutions available in class, students do not receive quality course content, and students are not given enough feedback about their programming skills. From the data in Figure [6], only 28% of students say they receive helpful feedback on their programming skills; 35% of students receive helpful feedback sometimes; and 36% of students are uncertain of their programming skills because they almost never or rarely receive helpful feedback. These statistics do not mean that instructors are not giving feedback at all. The question specifically asked students to rate the helpfulness of the feedback, and the majority of students do not consider the feedback received, if any, to be helpful. Moreover, students cannot always rely on the materials provided in class. Figure [4] shows that only 47% of students find the course materials easy to understand for someone who is still learning the subject. The remaining students say the materials are helpful sometimes, rarely, or almost never. Although Computer Science courses could be difficult, which means they require more studying on the student's behalf, Figures[16 – 18] show that Computer Science students are studious. The quality of the content makes an impact on students' ability to understand and apply the programming techniques effectively. As such, the convoluted content given adds more pressure on students.

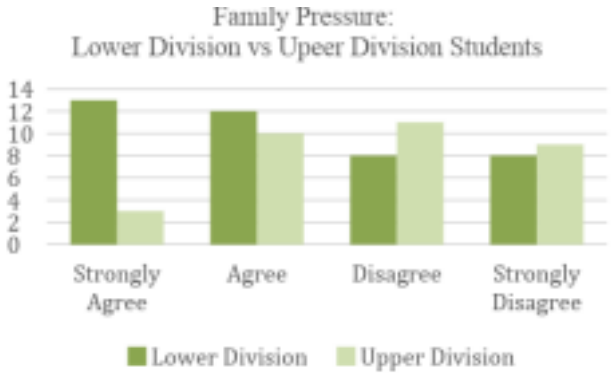
Connecting the analysis back to the framework discussed earlier that explains why cheating happens in Computer Science courses, the first component of the framework is pressure. In the case of the findings from this section, this pressure is due to the way some Computer Science instructors conduct their classes. Computer Science instructors must invest more time in providing quality content and feedback to lower the pressure students feel in Computer Science courses.

B. Students' Lifestyle

Students do not only experience pressure in class, but they also have other life circumstances that could play a role in how they conduct themselves academically. Social problems are probably difficult for instructors to solve; however, if instructors are aware, then at least they could rectify the pressure from in-class by providing quality content, solutions, and feedback to students.

Awareness of the lifestyle of students in Computer Science courses is crucial in preventing cheating because, usually, cheating is a decision a Computer Science student makes outside of class, as he/she works on a programming assignment. 51 % of students agree or strongly agree that their parents or family members pressure them to get higher grades. Students across all grade point averages and course levels report family pressure. More numbers of lower division students report stronger family pressure compared with upper division students, see Figure [29]. Family pressure is closely related to financial burdens. 64 % of students in Computer Science courses work on a regular basis to financially support their families, and 31 % share that they have personal circumstances which prevent them from studying to their best ability. These personal circumstances may include many factors, such as illness, homelessness, hunger, or other social problems. Data from this question is closely related to research by the Hope center which conducted a research with "more than 30,000 two- and 4-year college students [and found] that approximately half are food insecure, and recent estimates suggest that ... between 11% and 19% of 4-year students are housing insecure. Most of these students work and receive financial aid" [12]. These difficult living circumstances add more pressure on students.

Figure[29]



Yet, despite the rise in Academic Integrity violations and the high-pressure students face in and outside of class, students taking Computer Science courses study diligently. In terms of course hour commitment, 57% of all students take 15 to 17 college credits, 2 % take 11 or less college credits, 23 % take 12 to 14, and 15 % take 18 to 20 college credits. On top of the heavy course commitment, students study outside of class. On average, a student taking a Computer Science course sleeps 6.5 hours per day, eats 2 hours per day, and spends 5 hours per day studying and doing homework. However, there is a subset of students, specifically 27 %, which is nearly a third of students, who study 7 to 12 hours per day, with an average of 9 study hours per day. Also, students still find time to gain professional experience through internships and jobs. 22 % of students in Computer Science courses had at least one internship, and 64 % work to provide financial support for themselves and/or their families. Essentially, this is a focused group of students who want to succeed, and they have excellent skills to balance a college workload in addition to internships, work, and various life challenges. The pressure from life circumstances builds on top of in-class pressure, and this proves that the pressure component of the framework, which is used to explain cheating, is evident. Students become more vulnerable to breaking the Academic Integrity Policy when they are burdened by inadequate resources in class plus life circumstances.

An Academic Integrity Policy, which does not match the social norms, may be the last thing to cross the minds or hearts of students that are too focused on succeeding. The next section is dedicated to showing how the Academic Integrity Policy is not up to date with the culture of Computer Science students.

C. Students on the Academic Integrity Policy

Undergraduate computing students exhibit strong opinions on their competency, yet they are willing to consider violating the students' honor code under certain circumstances. Of the students surveyed in this research, 97% either strongly agree or agree that they can correctly explain a randomly selected programming assignment that they submitted for a previous class. Additionally, 97% say that they do not need to cheat in order to pass. If computing students are graded unfairly on a homework assignment, 90% disagree or strongly disagree that replicating the unfairness, by cheating, is acceptable. And, as seen in Figure [25], 56% of students say they almost never or rarely struggle with programming techniques in their current classes as they have practiced such techniques enough in earlier classes, 26 % say they struggle sometimes, and 17 % say they struggle often or almost always. The majority of students show assurance of their own technical abilities without deviating from academic integrity policies, but responses to more probing questions in the survey show different details on the students' attitude toward the Academic Integrity Policy.

From the 77 students surveyed, no one had a grade point average (GPA) lower than 2.5, and 23 students did not

have a grade point average as it was their first semester in college. 18.5 % of students often or almost always worked with groups on individual assignments, demonstrating an instance of excessive collaboration. Within this sub-sample of students who work in groups, all students have a GPA of 3.0 or higher. In fact, students with a GPA of 3.5 to 3.9 are the only group of students who almost always excessively collaborate on individual assignments. Although research, such as [11], shows that collaboration improves learning, group work on individual assignments is a clear form of cheating, per the Academic Integrity Policy [6]. However, not all students with high grades cheat. 42.5% of students with a 3.0 GPA or higher seldom or almost never cheat on individual assignments. The remaining 41% of students do not fall on either extreme; they only cheat "Sometimes." Students with a GPA of 2.5 to 2.9 form the minority in this study, as there are only 9% (7 students). Of these students, 6 of the 7 almost never cheat and 1 of 7 seldomly cheats. Due to the small number of students with low grade point averages, it is difficult to determine if there is a relationship between students who do not collaborate with others on individual assignments and students with a low-grade point average. However, it is obvious that unauthorized collaboration on individual programming assignments does not always equate to high grades. As such, a student contemplating cheating, with the false assumption that cheating yields high grades, should reconsider their decision. Students with high grades fall on both ends of the spectrum; there are students who violate

Student Honor Code and work with groups on individual assignments, and there are students who earn high grades without violating the Student Honor Code.

Not only are students divided about excessive collaboration on individual assignments, students in Computer Science courses have divided views about the use of resources in the presence of the Academic Integrity Policy. 45% of participants are willing to use any resources to learn the course content, even if the school policy explicitly prohibits the use of such resources. Of the freshmen participants, 45% of freshman reported that breaking the Honor Code is acceptable as long as they learn the information in the end. Similar to the freshmen class, 46% of seniors disregard school policies; they are also willing to use any online or in-person resources to complete their programming assignments. The number of freshmen and seniors who demonstrate willingness to violate the Academic Integrity Policy is closely related. Perhaps a student's belief of adhering to the Student Honor Code is not

changed throughout the four years of college. This point raises a concern on whether institutions effectively communicate the importance of Academic Integrity to students. As seen in Figure [21], 36 % of students share that taking the Computer Ethics course did not help her/him evaluate ethical conflicts in the Computer Science and Computer Engineering Discipline. Also, considering that 47% of students in this survey are either juniors, seniors, or master / 4+1 students, which is a large number of upper division students, 64%, still have not taken the Computer Ethics course. Lack of effective communication about the importance of Academic Integrity causes students to have disagreeing views about the use of resources in the presence of the Academic Integrity Policy, revealing that unfortunately, even if the school policy explicitly prohibits certain resources, 45% of students are willing to use these resources anyway.

A poll revealed the top three reasons why students in computer science and computer engineering courses do not believe that reading old programming assignments is an act of cheating, see Figure [22]. A total of 172 votes were received for the poll. Of the 172 votes, 55% categorized the act of obtaining and using assignments from a previous semester as non-cheating, 31% of the votes categorized the act as cheating, and 22% said the act was not honest. Not copying was the most common justification for not thinking of a behavior as cheating for students involved in cheating incidents. However, cheating is not limited to the act of copying. The Student Honor Code declares the act of possessing, reviewing, or using any materials intended to be used for any academic evaluation to be cheating [6]. The second and third most common justifications were that the best way to learn code is to read code, and that everyone looks up code

online, for 52.6% and 18.4% of the votes, respectively. Nonetheless, a third of the students admit that reading old programming assignments is a gateway to cheating. 30.3% agree that students who read leaked programming assignments cheat because they will likely write code similar to the leaked code. 26.3% say this behavior is a form of cheating as it can lead to copying, and copying violates the school policy. Only 14.5% of students believe that possessing, reviewing, and using leaked programming assignments is an offense worthy of an Academic Integrity violation. This data highlights a clash between students' belief on the definition of cheating and the Academic Integrity definition of cheating. An Academic Integrity Policy does not live up to the social norms of students taking

Computer Science classes.

Analysis shows that students rationalize the act of cheating with strong justifications, and this finding comprises the second component of the framework which gives a clarification to the underlying reasons of why students cheat. Students can freely justify cheating because the wording of the Academic Integrity Policy is too vague and diluted for a technical major like Computer Science. Thus, Computer Science students do not understand the boundaries of cheating, they have divided views about following the Academic Integrity Policy, and they do not always receive effective communication from the university about the importance of academic honesty.

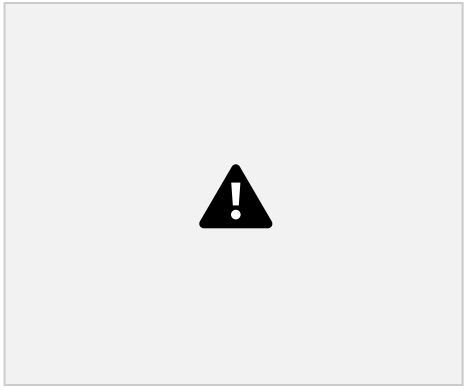
D. Students Evaluation of Themselves

The opportunities to violate the Academic Integrity Policy are difficult to eliminate. 93% of participants agree that learning the material is just as important or more important than earning high grades, so students can distinguish between earning grades and acquiring knowledge. Also, from the poll data, students have demonstrated incentives beyond acquiring a high GPA which motivate them to violate the Academic Integrity policy and use any venue to assist them

in completing their individual assignments, with disregard to the Academic Integrity Policy. Example of such venues include, but are not limited to, websites like Chegg, where students pay for solutions or individual tutoring; repositions like GitHub, where students find a copy of a source code for the same or similar assignments; and freelance markets like Upwork, where students pay a professional to complete their projects. Also, when students want, they have the opportunity to collaborate on individual assignments with each other. Instructors may not be able to catch students who collaborate without copying from each other even though, they disadvantage students who adhere to the Student

Honor Code and work alone on individual assignments. Nonetheless, the opportunities to cheat through paid help from online services or in-person collaboration are easily available for those who seek them.

In addition to the resources used for cheating, students allow each other the opportunity to cheat without being reported. This is mainly due envy; 39% of Computer Science students strongly agree or agree that they feel jealous of people who score higher grades than them. When students were asked why they do or do not report cheating, the most frequent explanations were that they either felt that the cheating behavior of another student does not impact them, or that they will not report the cheater so that the cheater can feel the punishment in the long term. The latter attitude is malicious. Not only does it allow the opportunity for cheaters to escape punishment, but it also opens the opportunity for other students to become more lenient about violating the Academic Integrity Policy. As was shared in the survey comments, students can feel disadvantaged for adhering to the school policy. Only 17% of respondents said "yes," they would report a cheater, see Figure [28]. The dynamic of students' feelings toward each other and the wide availability of cheating resources open the opportunity more cheating. This opportunity forms the third component of the framework, which is used as a comprehensive explanation of why cheating happens in Computer Science courses.



V. DISCUSSION

The epidemic of cheating stems from cultural clashes between university expectations and established norms among computing students. The survey results demonstrate that it is normal for nearly half of the students to collaborate in groups on individual assignments and use any sources as learning guides, even though these sources may not be in line with the Student Honor Code. The widely available resources include, but are not limited to, websites like Chegg, GitHub, Upwork, and many more that are easily accessible at the press of a button. These

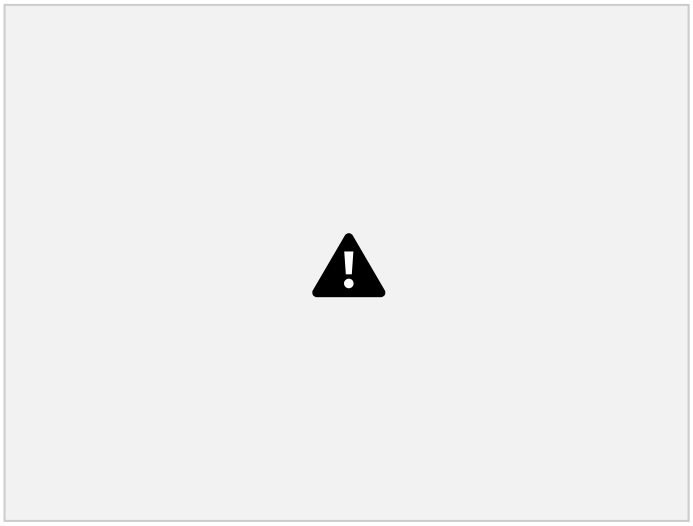
resources are too appealing for students not to use. However, the Academic Integrity Policy declares the acts of possessing, reviewing, or using any materials intended for academic evaluation as cheating. This is a result of a mismatch between university policies and behaviors that are widely accepted among students.

In addition to the availability of cheating resources, students face several pressuring factors inside and outside of school. In the classroom, nearly half of the students reported that their instructors do not share solutions on a consistent basis, even after the assignment is due, some instructors do not share the solution. As data demonstrated, students are more likely to utilize online resources than to seek additional help from instructors. Outside the classroom, the majority of students hold jobs to support themselves or their families. With financial burdens, students are under more pressure to succeed. In fact, nearly a quarter said their family pressures them to earn higher grades, and a third of students have personal circumstances which prevent them from studying to their best ability.

Students have also shown that they have justifiable reasons to commit a violation of the Student Honor Code. Although reading source code for a similar programming assignment from a previous semester is cheating, most students justified the behavior as not cheating. The most common justification for students who do not see this behavior as cheating was that the student involved in the incident did not copy. The second and third most common justification was that the best way to learn code is to read code. Nearly half of students have also justified that violating the Integrity Policy is acceptable in the process of learning. Students have shared sound explanation for cheating behaviors.

The pressure, justification, and opportunities students experience in Computer Science courses are the reason they cheat, and these three components form together a triangle analogous to the Fraud Triangle framework posed by criminologist Donald Cressey. In Cressey's Fraud Triangle Framework, the three elements "motivation, opportunity, rationalization" respectively correspond to the pressure students experience inside and outside of class, the availability

of resources to cheat, and the justifications students give about cheating behaviors [13]. The following is the Fraud Triangle.



Cressey's Fraud Triangle is mainly used to detect and prevent dishonest behaviour in a corporate setting; however, academic dishonesty is a fraudulent behavior similar to fraud in a business environment. In fact, a research study by [14] found that "students who cheat in high school and/or college are more likely to engage in certain deviant behaviors in the workplace," verifying that academic dishonesty in college translates to professional dishonesty in the workplace. Using Chegg to copy a homework solution, paying a professional for help on a programming project, or relying on source code posted on GitHub to complete an assignment are all factors that make the students vulnerable of committing fraudulent behavior in the workplace. [14] exposed that fraudulent behaviors which are likely to be committed by students who cheat include: overstating hours worked, surfing the web for personal use during working hours, overstating expenses for reimbursement, stealing at work, and being slow and sloppy in finishing work tasks. Cheating in Computer Science courses is a form of a fraudulent behaviour per Cressey's Fraud Triangle Framework. Like employees who commit fraudulent behavior in the workplace, students in Computer Science courses have the opportunity to cheat, feel pressure from family and school, and justify certain cheating acts as norms.

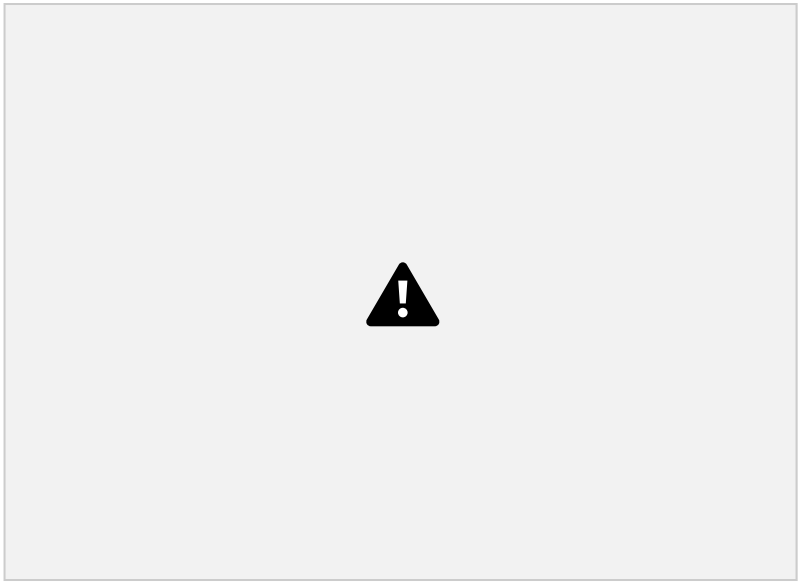
Instructors and administrators could benefit from utilizing the Fraud Triangle in an academic setting. The Fraud Triangle can be used to understand the variety of reasons which motivate students to cheat. Understanding, detecting, and making solutions to various cheating incidents becomes easier now that there is a framework constructing a full

picture of why students in Computer Science courses cheat. Motives which form the components of the Fraud Triangle can be addressed individually, and one at a time, to help guide students into academic honesty. Ensuring that students

embrace and maintain academic honesty in college is an indicator that they will have academic honesty in the workplace, as discussed earlier. The Fraud Triangle framework is a valuable asset in helping administrators and instructors support Computer Science students.

VI. PROPOSED SOLUTIONS

This section is dedicated to highlighting solutions on ways to overcome the Fraud Triangle in Computer Science courses. As [15] described, overcoming the Fraud Triangle is possible by developing controls, investing in culture, and upholding values that stand against opportunity, pressure, rationalization.



A. Develop Control to Lower Opportunity

- ❏ Write an Academic Integrity Policy specific to the Computer Science program
- ❏ As seen from the survey comments, students do not report cheating because they do not know how to report it in a confidential manner. Make a tool available and known to Computer Science students, in which they can report cheating incidents, and upload code files as evidence of the cheating incidents. It is preferred to keep the report confidential, to lower issues of retaliations

B. Invest in Culture to Lower Pressure

- ❏ Devote more time in providing quality content and feedback to lower the pressure students feel in Computer Science courses
- ❏ Implement measures to ensure students receive quality content in class; in addition to adequate, comprehensive, and clean code examples and/or solutions

C. Uphold Values to Lower Rationalization

- ❏ Ensure Computer Science students complete the Computer Ethics course early in the program, by making the Computer Ethics course a prerequisite to an upper division course, such as Capstone. Also, stimulate Computer Science students with discussions about academic honesty, discuss the boundaries, issues, and outcomes of cheating, to help students be honest in a Computer Ethics course or in future courses
- ❏ In Computer Science courses, instructors may permit students to use a "cheat sheet" during exams. Perhaps, the term "cheat" can be eliminated from the word "cheat sheet," as it may impact the unconscious mind of students into thinking that cheating is normal. Alternative words to "cheat sheet" can be "exam sheet" or "note sheet." As [16] said "Language is powerful; improperly used, it can confuse and obfuscate. Properly used, it can clarify and illuminate."

VII. CONCLUSION

The goal of this study is to equip administrators and instructors with a deeper understanding of the apparent cheating problem in Computer Science courses. Survey results were gathered to collect both numerical and textual data about students' perspective of instructors, the academic integrity policy, and each other. Questions about students' lifestyle were used to give a deeper understanding of students' background. The analysis section guided the reader through the formation of the Fraud Triangle framework, which gives a modular answer to why students cheat, finding pressure, opportunity, and justification as the three components. This Fraud Triangle framework is used in a professional environment to understand fraud at the workplace, and this research demonstrates that the fraud triangle is a valuable tool when addressing cheating in Computer Science courses. After discussion of the Fraud Triangle,

some

proposed solutions highlighted strategies that administrators and instructors can implement to increase academic honesty in computer science courses, with respect to the Fraud Triangle and comments received from students in the survey. With a strong understanding of the cheating problem in Computer Science courses, administrators and instructors will also be able to think of more effective solutions to increase academic honesty among students.

VIII. ACKNOWLEDGMENT

First and foremost, I would like to thank God, the most merciful, for giving me the strength, knowledge, ability and opportunity to persevere and complete this research study satisfactorily. Without God's blessings, this achievement would not have been possible.

I have great pleasure in acknowledging my gratitude to my thesis director, Professor Navabi. She has been there providing her heartfelt support, invaluable guidance, inspiration, and suggestions in my quest of writing my thesis. I

would also like to express my gratitude to Professor Bazzi, who has given me the freedom to shape my research, while silently and non-obtrusively ensuring that I stay on course.

My acknowledgement would be incomplete without thanking my family. The blessings of my parents, Mrs. Awamleh and Mr. Alyasari, and the love and care of my sister Zahraa and brother Furat have made a tremendous contribution in helping me finish my thesis. I also would like to thank my dear friends Amirah and Nema for being so encouraging.

I would like to dedicate this work to all students and instructors of Computer Science courses. We all can benefit by discussing a problem that affects all of us!

- Farah Alyasari

IX. REFERENCES

[1] J. Bidgood and J. Merrill, "As Computer Coding Classes Swell, So Does Cheating", *Nytimes.com*, 2017. [Online]. Available: https://www.nytimes.com/2017/05/29/us/computer-science-cheating.html.

[2] C. Marsan, "Why computer science students cheat", *Network World*, 2010. [Online]. Available: https://www.networkworld.com/article/2207189/why-computer-science-students-cheat.html.

[3] K. Huangpu, "Competitive environment drives culture of cheating in computer science classes", Columbia Daily

Spectator, 2018. [Online]. Available: https://www.columbiaspectator.com/news/2018/02/22/competitive-environment drives-culture-of-cheating-in-computer-science-classes/

[4] E. Indani, "The CIS department has the most cases of academic dishonesty — but not for the reason you think", Thedp.com, 2017. [Online]. Available: https://www.thedp.com/article/2017/10/the-cis-department-has-the-most cases-of-academic-dishonesty-but-not-for-the-reason-you-think

[5] A. Lathrop and K. Foss, Guiding students from cheating and plagiarism to honesty and integrity: strategies for change. Westport, CT: Libraries Unlimited, 2005.

[6] Provost.asu.edu. (n.d.). *Academic Integrity at ASU | Office of the University Provost*. [online] Available at: https://provost.asu.edu/academic-integrity.

[7] W. D. Roth and J. D. Mehta, "The Rashomon Effect," *Sociological Methods & Research*, vol. 31, no. 2, pp. 131– 173, Nov. 2002.

[8] S. Kistler, "Susan Kistler," AEA365 Likert scale Tag, 27-Dec-2012. [Online]. Available: https://aea365.org/blog/jan-losby-and-anne-wetmore-likert-scales-odd-or-even/

[9] "'even vs. odd point scales,'" Insight Central. insights for maximum marketing performance, 14-Jul-2010. [Online]. Available: https://analysights.wordpress.com/tag/even-vs-odd-point-scales/

[10] E. Singer and M. Couper, "Some Methodological Uses of Responses to Open Questions and Other Verbatim Comments in Quantitative Surveys", pp. 1-19, 2017. Available: https://www.gesis.org/fileadmin/upload/forschung/publikationen/zeitschriften/mda/online_first/mda_Singer.pdf.

[11] Haycock, K. (2007). Collaboration: Critical success factors for student learning. *School Libraries Worldwide*, [online] 3(1), pp.25-35. Available at: https://scholarworks.sjsu.edu/slis_pub/24/

[12] Broton, K. and Goldrick-Rab, S. (2017). Going Without: An Exploration of Food and Housing Insecurity Among Undergraduates. *Educational Researcher*, 47(2), pp.121-133.

[13] Schuchter, A. and Levi, M. (2013). The Fraud Triangle revisited. Security Journal, [online] 29(2), pp.107-121.

Available at: https://link.springer.com/article/10.1057/sj.2013.1

[14] Graves, S. and Stephen, F. (2008). Student Cheating Habits: A Predictor Of Workplace Deviance. Journal of Diversity Management – First Quarter 2008, 3(1), pp.15-19.

[15] “White Paper: The Anti-Fraud Triangle,” Radical Compliance, 14-Sep-2017. [Online]. Available: http://www.radicalcompliance.com/2017/09/14/promotion-anti-fraud-triangle-paper/.

[16] Field-Fote, Edelle PT, “https://journals.lww.com/jnpt/fulltext/2009/06000/The_Power_of_Words.1.aspx,” Journal of Neurologic Physical Therapy, vol. 33, no. 2 - p 67, 2009.