# Program-Specific Report Deliverable

Katie Houskeeper, Morgan Hughes & Addie Suckow
Worcester Polytechnic Institute
2022 UIR ABET Accreditation IQP Team
gr-uir-abet-d22@wpi.edu

---

Our team created this Program-Specific Report as a deliverable within our project, Evaluating Program Compliance with ABET Standards within the College of Engineering at the International University of Rabat (UIR). This project fulfills the Interactive Qualifying Project (IQP) Degree Requirement of our university, Worcester Polytechnic Institute.

Our team has compiled findings in both words and visuals as well as recommendations for the UIR College of Engineering's Computer Science program. These findings were created based on data collected from direct and indirect evaluations of the program. We conducted a direct evaluation by analyzing course syllabi, which included a general examination of formatting, completeness, and contradictions within them as well as a more in-depth assessment of its alignment of course outcomes and student outcomes. We completed an indirect evaluation of the program by creating and distributing surveys as well as conducting interviews. These were carried out on four UIR stakeholder groups: students, faculty, alumni, and employers.

Implementing or further investigating these recommendations will propel the Computer Science program toward better alignment with ABET Accreditation standards.

# Table of Contents

# Introduction

Our team completed the second initial accreditation review for the Computer Science program. We believe that this program has many positive aspects, but there are also many areas that need to be improved to align with ABET standards. A majority of these are laid out in the following sections.

# Lack of ABET Understanding

It was important for our team to get an idea of faculty's understanding of ABET accreditation and their feelings regarding the program pursuing it. We aimed to do this in our interviews, but due to the lack of faculty that were willing to participate, we were unable to gain a holistic understanding. With this in mind, our team believes there may be opposition from faculty regarding pursuing accreditation. It is important for all faculty and staff to thoroughly understand ABET and the benefits of accreditation before they are expected to submit to the evaluation process and submit all the necessary materials.

The faculty's lack of understanding will greatly impact the willingness to adopt any other recommendations made for the program, so a precedence should be set on addressing it first. Our team recommends explaining ABET and its benefits to faculty members, hearing their concerns, and answering their questions about accreditation and its processes before expecting them to support all the changes. This implementation could take time, but seminars on the accreditation efforts, infographics, or other means of conveying this information could be greatly beneficial in continuing the efforts of improvement.

Once faculty members are well-versed on what the accreditation process entails and the direct benefits they will see in the future, our team believes that opportunities to practice course outcome creation and syllabus development in the French language would be well-received. We offer this suggestion in hopes of increasing faculty's understanding of ABET and, in turn, their willingness to adapt their course structure and methods of assessment.

# Course Outcome and Student Outcome Alignment

A major finding that our team came across when analyzing the Computer Science program syllabi was the errors regarding course and student outcome alignment. Specifically, many of the course outcomes do not reflect every aspect of a course as denoted in the syllabus. Often, there were areas that a course should have aligned with multiple student outcomes, but the verbiage in the course outcome (CO) didn't show the proper alignment. This was especially prevalent for student outcomes (SOs) 4 and 5. Figure 1 below,

shows this uneven distribution of SOs throughout the Computer Science program. As is with all tree maps, the size of the box indicates the frequency of alignment.

It is important to note that we were only given seven syllabi from the Computer Science program. This means that all of the data that will follow regarding the syllabi findings is not representative of the entire program. Due to this, we recommend that a more holistic review of all the course syllabi is done in the future, so there is data that represents the entire program.

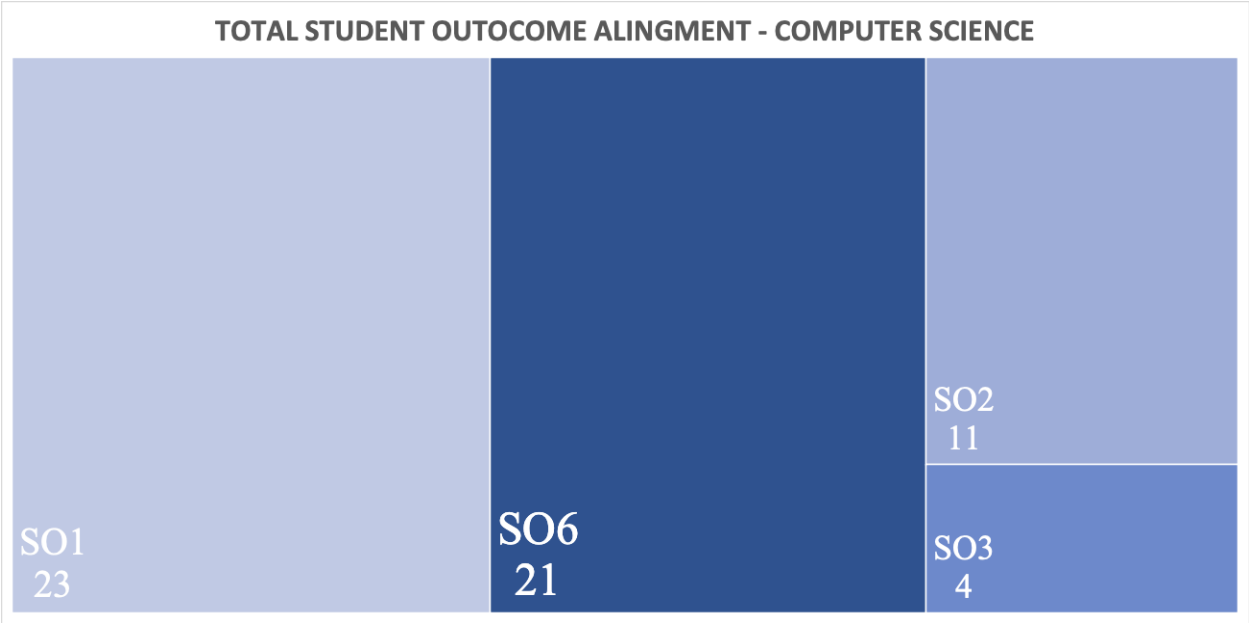



*Figure 1: Total Student Outcome Alignment within the Computer Science Program*

This visual shows how a majority of COs align with Student Outcomes 1 and 6, and very few align with the other SOs. Student Outcome 1 aligns with 23 of the 27 course outcomes, which amounts to 85.19% of all COs. Comparing this to the other SOs the imbalance is clear, especially when looking at Student Outcomes 4 and 5 which do not align with a single course outcome from any of the syllabi we were given.

When analyzing the syllabi, there were many instances of discrepancies between the course outcomes and the topics outlined on the syllabus, as mentioned earlier. While this is seemingly irrelevant, it directly makes the issue of uneven distribution more drastic. It is very important to ensure that all aspects of a course are mentioned in the course outcomes. This makes sure that wherever there is alignment of COs and SOs, the mapping demonstrates it. In the Formal Syllabi Notes, our team flagged multiple syllabi that needed to verify if teamwork was a component of the course, whether this be in lab experiments, group projects and assignments, or any other area. None of these courses aligned with Student Outcome 5

(which is involving teamwork), but there was some mention of it in the syllabi. This was a similar problem with SO 4 (involving ethical judgment), where there was once again no alignment.

The Formal Syllabi Notes document for the Computer Science program should be utilized to fix these errors. In order to correct these course and student outcome alignment issues, the course outcomes must be rewritten in each individual syllabus to ensure that courses are focusing on more than just SOs 1 and 6. The document provides specific phrasing suggestions that fully encapsulate the components of a course within its COs.

One thing that stood out positively from the Computer Science syllabi, was that they had the appropriate amount of course outcomes, which is considered to be between two and five COs. All the syllabi we were given had five or less outcomes, which is satisfactory. An evaluation of the rest of the syllabi should be done to ensure this is consistent throughout the remainder of courses within the program.

# Student Feelings Toward ABET Student Outcomes

Our team aimed to find out how current Computer Science students feel their abilities align with student outcomes through interviews and surveys. In the survey we asked this for each individual outcome, and the answers varied.

Figure 2 below looks specifically at Student Outcome 5, which is based around teamwork and group dynamic problem solving. In this category, a majority of students feel that their teamwork abilities are adequate or better, as shown visually below.

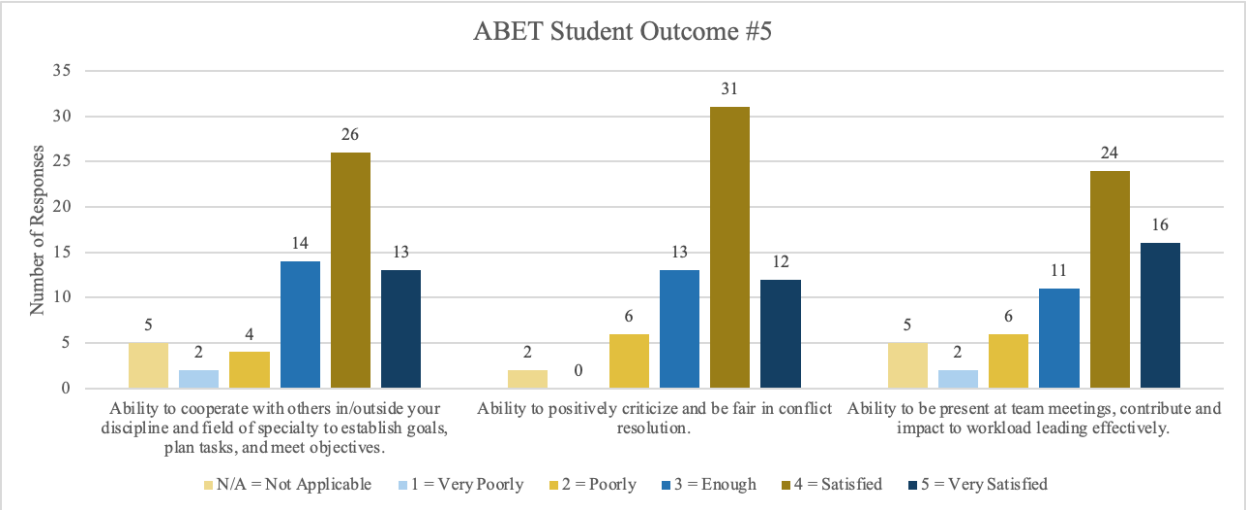


*Figure 2: Computer Science Student Survey Responses to Questions Regarding ABET Student Outcome #5*

When our team analyzed the alignment of course outcomes with these student outcomes, our findings did not back up what the students said in the survey. No course outcomes aligned with Student Outcome 5 within the Computer Science programs. In this case, we believe this ties back to the issue of not showing alignment within the syllabi. If students feel confident in their teamwork abilities, they must have more experience with it in their courses than indicated. This once again brings back the importance of ensuring that course outcomes display every aspect of a course and every point of alignment with SOs.

Even with the limited number of syllabi provided to our team by the Computer Science program, it can be inferred from survey data that instructors are likely not aligning their contents within their course to ABET standards. For example, as shown in Figure 3, many Computer Science students indicated that Student Outcome 2 was "Not Applicable" to them. When the student deemed it applicable, a rating of "Enough" was often given. All Student Outcomes should be applicable to students because it is a mandatory standard of ABET accreditation for students to display all outcomes upon graduation. This once again ties back to the issue of uneven distribution of student outcome alignment. It is important for courses to align with ABET standards, which includes the student outcomes.

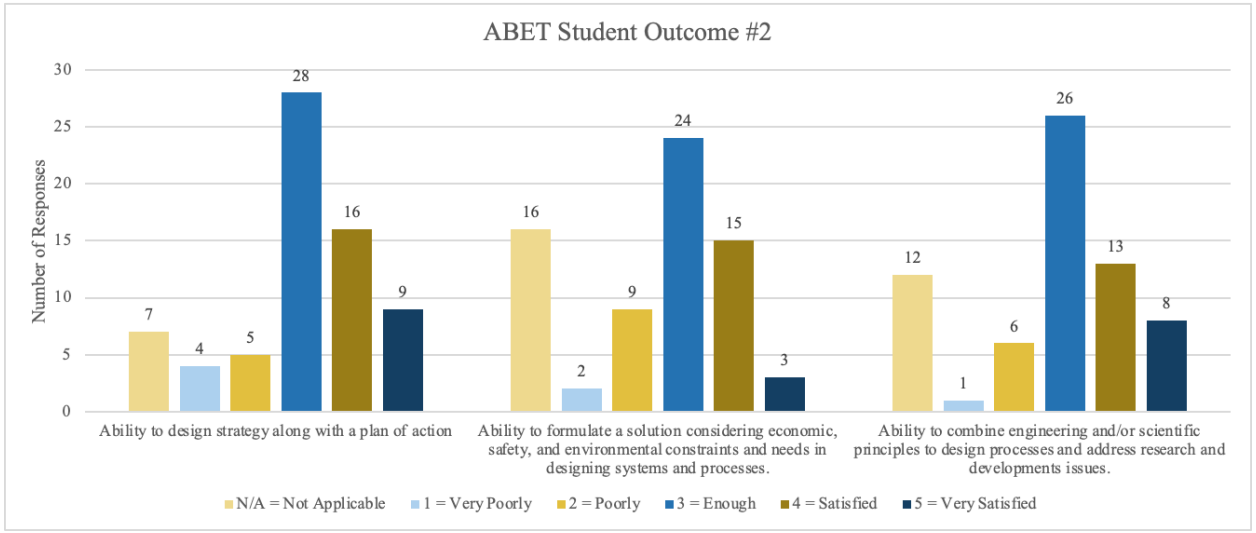



*Figure 3: Computer Science Student Survey Responses to Questions Regarding ABET Student Outcome 2*

# IRE System

Throughout our analysis of the syllabi and course outcomes, there were many findings relating to the IRE system. There is a small blurb on the bottom of the syllabi below course mapping where IRE is broken down to give professors a better idea of how to use it when mapping their COs, but the explanation itself is incorrect. On the syllabi it is laid out as seen below:

**I**= Introduce (Weak), **R**= Reinforce (Intermediate), **E**= Emphasize (Strong)

This explanation does not accurately describe the levels of IRE due to the connotation of each level with a degree of "Strong" or "Weak". Course outcomes that Introduce topics and align with the I level are not weak COs, they are simply introducing a subject using I level verbs. Students are expected to understand these topics and describe them, but nothing more. It is important to have I level COs in classes that are introducing topics as most disciplines build on the information students are taught previously.

At times throughout the syllabi our team was given, course outcomes did not use IRE verbs, which made it impossible for us to map their alignment with ABET's Student Outcomes. An example of this is shown below in Figure 4. As it displays, the original course outcome uses the word "master" which is not an IRE verb. Below is a suggestion to replace this word with the verb "understand" to better align with IRE and clarify the overall message of the CO.

**CO3: "Master** the basic concepts of OOP"

↓

**"Understand the basic concepts of OOP"**

*Figure 4: Adding IRE Verbs to Course Outcomes Example*

There are multiple variations of this happening throughout the course outcomes within the Computer Science syllabi that we were given, and these errors should be addressed.

Due to the limited amount of syllabi our team was given to evaluate, we were unable to conduct an analysis of a yearly IRE breakdown. In future self-studies, this should be calculated to determine if there is the proper distribution of I, R, and E throughout the five years of the program. Figure 5 below demonstrates what an ideal IRE breakdown looks like so when a future distribution is calculated, it can be compared to this chart.

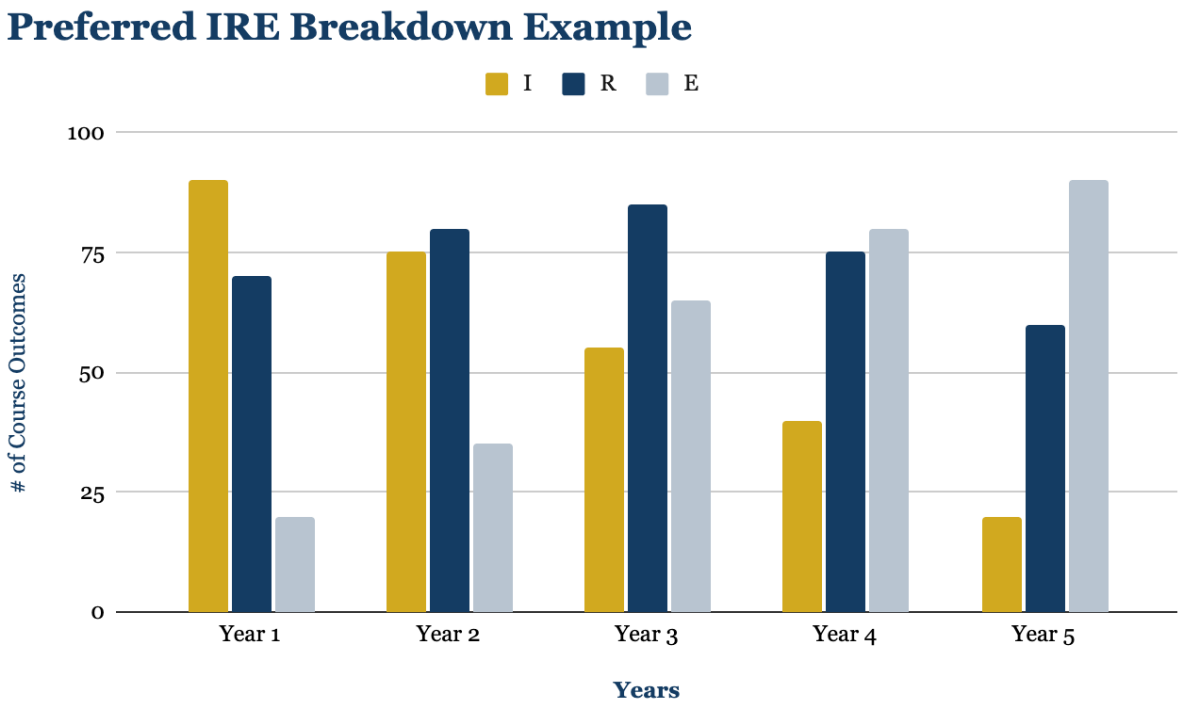



*Figure 5: Preferred IRE Breakdown Example*

As is shown above, I's, R's, and E's should be distributed throughout the years of a program. Level I should be largely used in the first two years, steadily decreasing as courses get more major-specific and complex. Level R should begin slightly lower than I, with a slight peak in the third year and marginally decreasing throughout the remainder of the program. Finally Level E should be used very rarely in the first two years, steadily increasing to be heavily used in the final year. This is important to address because it is something that ABET will specifically look for and is an important aspect of achieving accreditation.

The comprehensive list of IRE verbs our team used was included as a deliverable of our final report, and details the specific verbs and their correlation to Bloom's Taxonomy and the IRE system. It is our recommendation that the IRE verb list be distributed to all faculty members along with guidelines to follow when utilizing the list. It is important to not only give faculty the list, but also give them instructions on how to implement them in their course outcomes, because only doing the first step will not solve the issues of incorrect verbs.

Our team also recommends that the  explanation of I, R, and E on the syllabus should be changed to remove “Strong,” “Intermediate,” and “Weak” while keeping the meaning of IRE. Below is what it should be changed to.

**I**= Introduce, **R**= Reinforce, **E**= Emphasize

In order to better align with ABET standards, our team recommends looking at the verbs used in the course outcomes and making sure they are the appropriate IRE level. Focusing on starting with a majority of topics introduced in the first and second years, then steadily decreasing through the remainder of the program would better represent the goals of the program. Doing the opposite for emphasizing, starting lower than the other two levels and steadily increasing through the fifth year, would accomplish the same. This is another instance where the Formal Syllabi Notes are beneficial to use to help professors choose the appropriate verbs and rewrite their course outcomes accurately.

Ultimately, there are many areas of improvement within the Course and Student Outcome alignment section of the syllabus. A majority of these can be remedied by utilizing the Formal Syllabi Notes deliverable that is discussed throughout this section. Taking into consideration the notes and suggestions made within that document will benefit the professors and lead to more concise and correct course outcomes, along with better alignment with ABET Student Outcomes.

# Lack of Syllabi Consistency

Through an analysis of the general contents within each syllabus, our team found that, while the Computer Science program has a standardized syllabus, instructors lack a cohesive understanding of what content belongs in each outlined subsection.

Often, the syllabi lacked basic information such as the course ID, credit hours, or the prerequisites required to succeed in the course. The lack of basic syllabi components indicates that faculty members are not dedicating enough time to creating their syllabi or do not have enough time/resources to do so. Additionally, this finding led our team to believe that syllabi are not returned to course instructors and supervisors and directors do not provide professors with feedback on how to improve them. If instructors are given feedback on their syllabi, our team found that it is not edited and the existing, incomplete document is dispersed to students. Furthermore, the same mistakes will likely be made in the development of future syllabi as the feedback was not provided or not reviewed.

The Catalog Description included in the syllabi should offer a general overview of the topics and concepts covered within the course. However, we often came across descriptions that were incredibly vague or were not written at all. It should be noted that catalog descriptions can likely be recycled for that course once they are created and instructors would not need to create new ones every semester.

Based on all the findings described above, our team recommends providing program-specific workshops for faculty members where they learn how to correctly create a course syllabus using the template provided to them. In this workshop, faculty should be given the opportunity to learn what belongs in each section, practice writing and formatting the sections, and given exemplary syllabi to reference and model as they create the documents on their own.

Time should be spent walking faculty through each step to ensure that mapping is done correctly. It is also crucial to give faculty the time and space to ask any questions they have about mapping so they completely understand the need for it and its benefits. Overall, it would be beneficial to create a condensed guide on how to map COs properly so that professors can have the guide and syllabus side by side while they are mapping.

We also recommend that these faculty workshops are held in both French and English. This is due to the fact that, as mentioned previously, English is not the first language for professors. Giving them an understanding of the student outcomes in French first will ensure that they comprehend all aspects of the outcomes and how to align COs with them, which they can then translate into English on their syllabi.

Most importantly, we recommend that the timeline of syllabus creation and collection be altered with the intention of providing feedback to faculty members about what was done incorrectly on their syllabus and how to fix it. This should be followed up with the expectation of editing the syllabus until it meets the needs expected within each program. Essentially, our team suggests that syllabi are collected a few weeks prior to the start of a semester and reviewed by the instructor's supervisor. Then, it is returned to the instructor and they are asked to submit a revised version within a set deadline, before the semester begins.

We make these recommendations in hopes of furthering the benefits of the standard syllabus created last year. While the template exists, it is not currently achieving all it set out to because faculty members do not have standardized directions on the content nor a cohesive understanding of CO/SO mapping. The adoption of these recommendations will benefit the students within each course, as well as the faculty, administration, and future accreditation evaluators.

# Need for Greater Maintenance of Technology

While UIR's Computer Science department exerts significant efforts to provide faculty and students with innovative machinery and technology, the program must not overlook the importance of maintaining more basic technologies such as those within the four walls of a classroom. Students, faculty, and alumni alike indicated that malfunctioning classroom technology, such as projectors, often diverted time that could have been spent learning course material.

A common theme for all UIR students interviewed was that they appreciate the accessibility of technology infrastructure on campus, including computer labs. Students within the Computer Science program, in particular, indicated having the software and tools required for courses is beneficial as without the establishment of these computer labs, they would have to download, manage, and potentially obtain licensing for softwares on their own personal devices. However, in these areas, inconveniences are widely prevalent as keyboards and other computer accessories are sometimes missing, damaged, or faulty. In addition, operating system crashes have occurred, which is incredibly inconvenient, particularly in the instance of one student interviewed, who shared with us that it occurred during a midterm exam.

Our team recommends the Computer Science program place a high emphasis on investigating and correcting current technologies, including those within a classroom and laboratories and the undependable wifi. Once the program has the resources to maintain its existing equipment, we suggest they begin to fund the employment of new instructors within the Cybersecurity specialty as well as to reevaluate the current curriculum of teaching students Java, Matlab, and Python.

# Software Skills

By a student's fifth year in the program, they are expected to have strong abilities in several software skills that align with those utilized in industry. This idea is reinforced as students indicated they have a firm grasp of HTML/CSS/JS and R in Figure 6. However, for software including Java, Matlab, and Python, there are a startlingly high number of "Very Poor" and "Poor" responses. Affirming this finding, within faculty interviews, Python was noted as a software that is poorly understood and utilized by students.

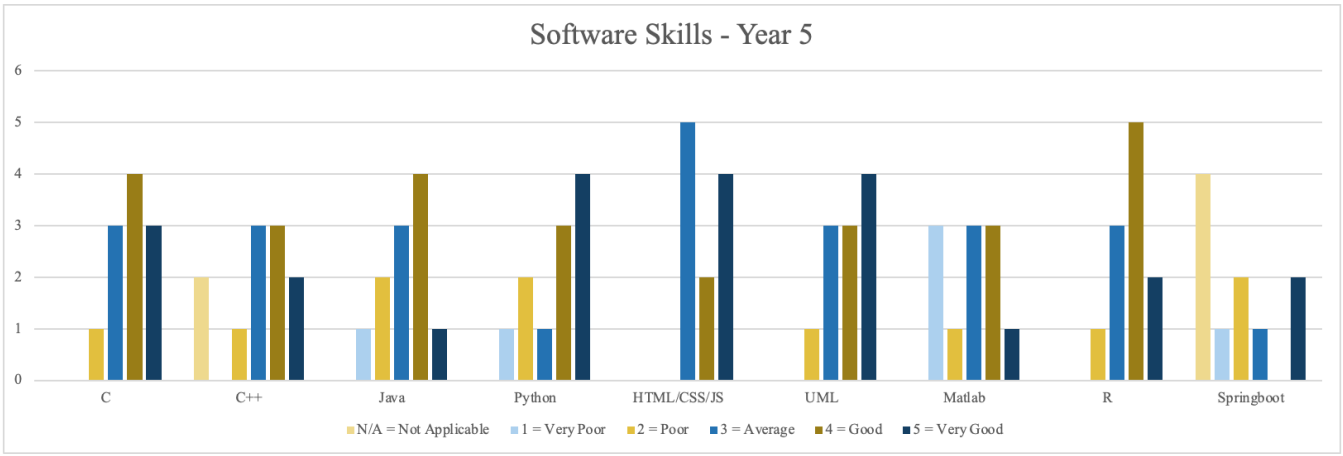



*Figure 6: Computer Science Student Survey Responses to Questions Regarding Software Skills*

# Laboratories and Practical Learning Methods

Figure 7 below indicates how students within all five years at the university rate their academic facilities. As discussed, computer facilities need improvements to better student experiences. As incorporating laboratories within a program is a requirement within ABET standard, 36% of students should not indicate laboratories are not applicable within their courses. An additional 15% of responses stated laboratory quality was “Very Poor” or “Poor,” which is troublesome.

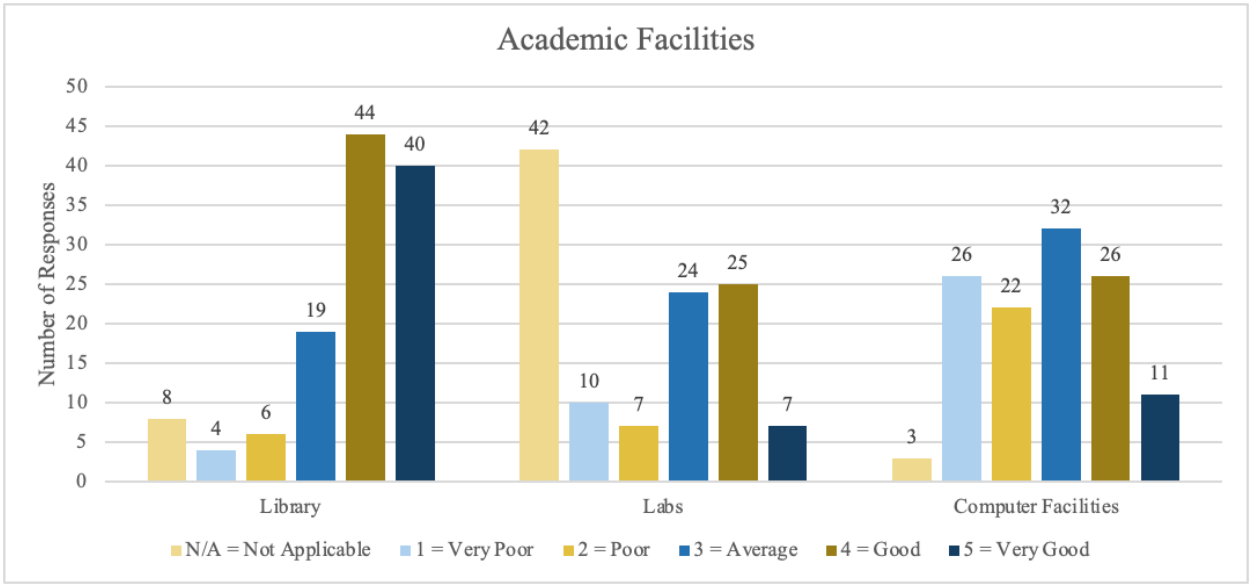



*Figure 7: Computer Science Student Survey Responses to Questions Regarding Academic Facilities*

Laboratories, especially those in which students can employ hands-learning methods, partition the lecture-heavy nature of school days. Students, especially those within the latter years of the Computer Science program, express concern with their ability to practice the theoretical topics they are learning in the classroom in live experiments or laboratory demonstrations. There appears to be excellent equipment within the Technical Hall and the Engineering building; however, there are not many opportunities to use the machinery. Alumni and students who had held internships expressed to us during their interviews that they entered the industry without portions of technical knowledge employers expect them to know. This is in part due to the limited number of labs and hands-on projects that were assigned to them while at UIR. For Cybersecurity laboratories in particular, faculty and students indicated a lack of instructors, proper licensing, and firewalls to provide protection for the network and computer user.

We also believe that the Computer Science program could greatly benefit from the addition of more lab components and hands-on work, specifically for fourth and fifth-year students. Citing the satisfaction of putting classroom theories into practice, increasing self-efficacy, and ease of learning through dynamic and interactive concepts compared to theoretics, students' hands-on and research-oriented work adds excitement and deviation from workload-heavy days. Often, priority is given to the traditional technical skills within a field. Once students have mastered those foundational competencies, they should be exposed to new ones. When creating new opportunities, there should be an emphasis on implementing the most recent advances in technology and the inclusion of more contemporary techniques needed in the current industry. This will allow for the refinement of both technical and soft skills, including collaboration techniques, learning how to manage projects strategically, and practicing leadership. Opportunities like these ensure that students know more than just the basic skills needed for industry and make them more competitive in the post-graduation job market.

# Industry Exposure

Along with the misalignment of practical methods and industry, our team also found that UIR Computer Science students and alumni feel they lack exposure to industry professionals overall. Throughout interviews, these students stated that university personnel did not provide opportunities to interact with employers and those working in industry. As shown in Figure 8, 73% of respondents within the computer science program indicate UIR faculty did not help them locate career-related opportunities. Additionally, shown in Figure 9, 60% of those respondents also indicated while at UIR, they did not even have the chance to interact with potential employers.

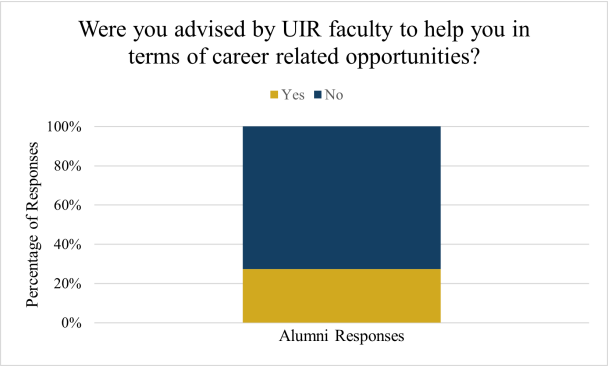



*Figure 8: Computer Science Alumni Survey Responses to Questions Regarding Career Advising at UIR*

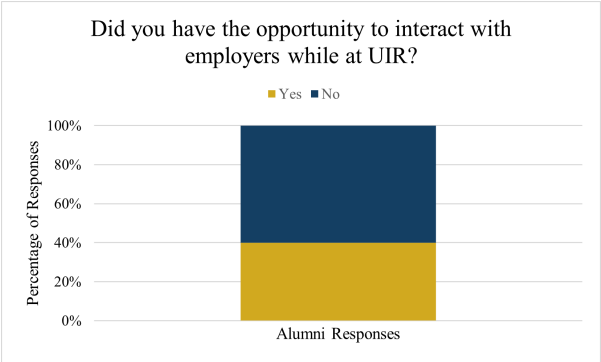


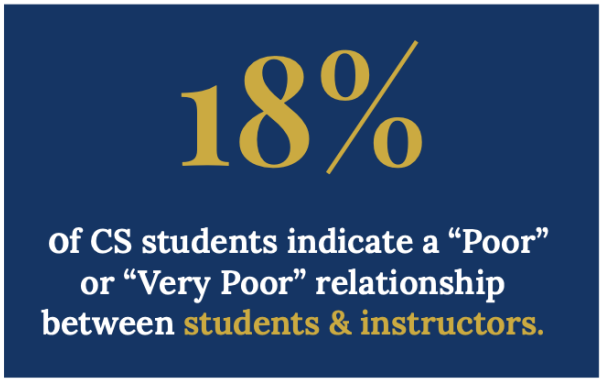

*Figure 9: Computer Science Alumni Survey Responses to Questions Regarding Interaction with Employers*

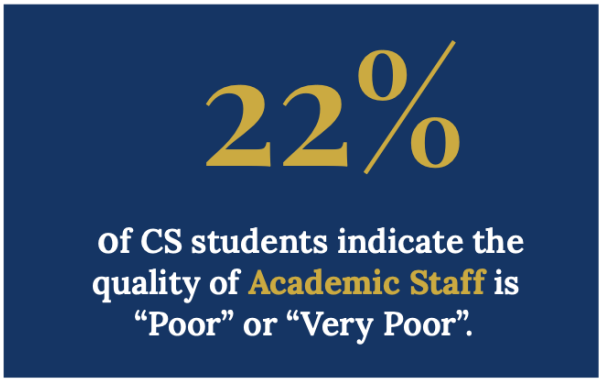
With the countless number of strong and sustainable academic and industrial partnerships that the Computer Science program maintains, our team believes there is a missed opportunity for students to connect with these individuals. There is great value in networking and having an inside perspective into what is currently happening in a student's intended field. As a team, we recommend the program put more effort into promoting the avenues they have currently established for career development, such as providing students increased visibility to the Office of Student Life (Vie Estudiantine). This department could also host workshops, such as resume and cover letter reviews, that would aid students in creating and compiling high-quality materials that can prepare them to enter the job market. Additionally, it would be beneficial to have a single database that can house all contact information for the companies that have employed students from the Computer Science program. This would be a resource that could lead students in the right direction as they begin their search for post-graduate opportunities. Similarly, the program could compile an alumni list which would allow students to seek advice from an individual who was once sitting in the same seats they are now.

Lastly, program-specific career fairs could be another great avenue for the program to investigate. These events could provide students more industry exposure and information about post-graduate opportunities, while allowing them to practice professional communication. Additionally, this event would provide visibility to the program and potentially establish partnerships with various companies interested in hiring students. These partnerships would provide long-term benefits for current students and increase the interest from prospective students.

# Student Relationships with Campus Personnel

When surveyed regarding the quality of their relationships with various members of academic personnel, our team found an increased amount of Computer Science students having “poor” or “very poor” relationships with instructors and academic staff alike in comparison to other College of Engineering programs.




Although professor-student relationships may seem unimportant to an unknowing party, they can substantially impact student and teacher performance. Students who have strong relationships with their instructors often report they are more motivated to complete their work to a high quality and feel a greater willingness to ask for help when needed. On the contrary, when students feel they are not receiving the quality of teaching they deserve, they are not as determined to fully apply themselves. Due to this, our team recommends the Computer Science program further investigates student-professor relationships and identify areas of improvement.

# Conclusion

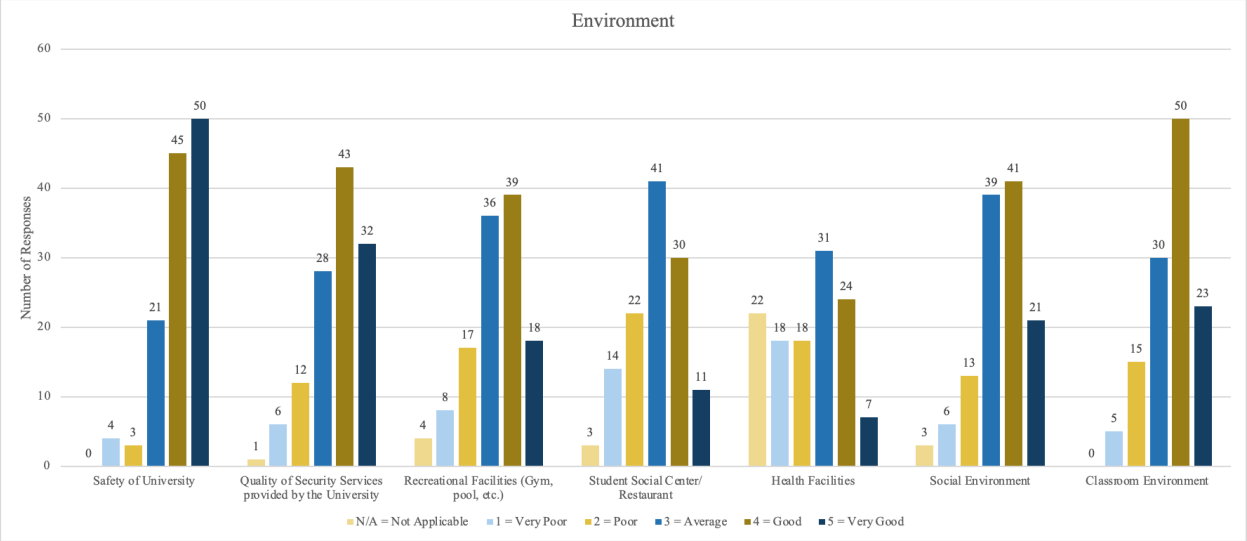
As the Computer Science program was only able to provide our team with seven syllabi to review, before seeking an official ABET Accreditation, our team recommends using the tools we have provided as

deliverables to complete another direct evaluation of their course syllabi. Additionally, faculty will need to be willing to put in a more significant amount of effort and support into this endeavor as very few interviewed with our team.

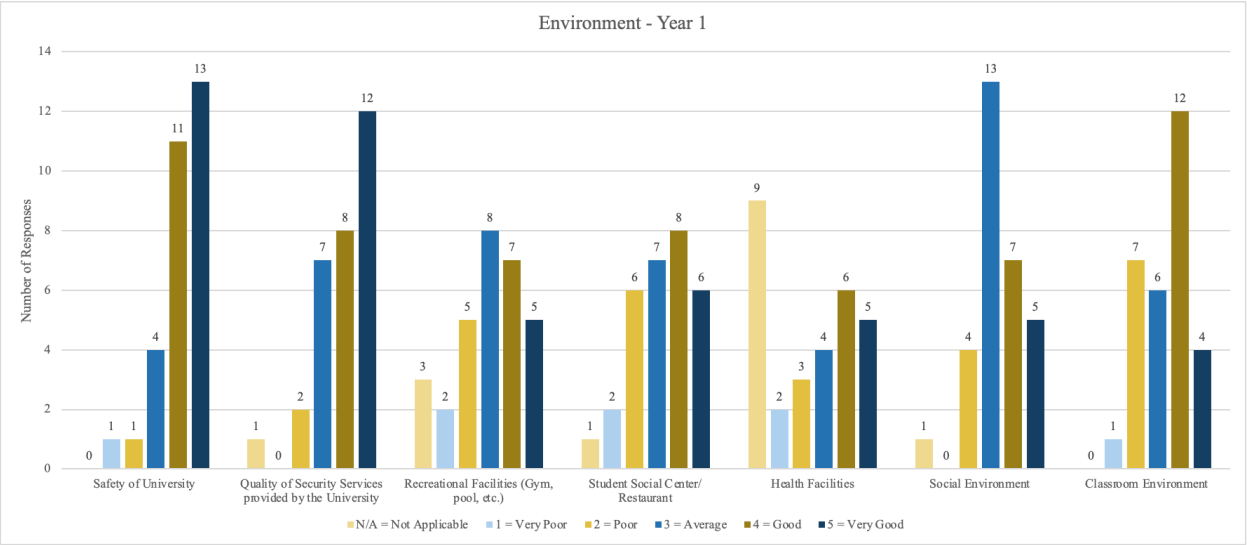
# Appendix A. Student Survey Responses

Included below are all visual representations of the data we received from Computer Science programs students on a survey distributed in late March. The raw data has been shared with the Computer Science program. In this appendix you will find each section of survey questions along with the total responses as well as responses filtered by class year.

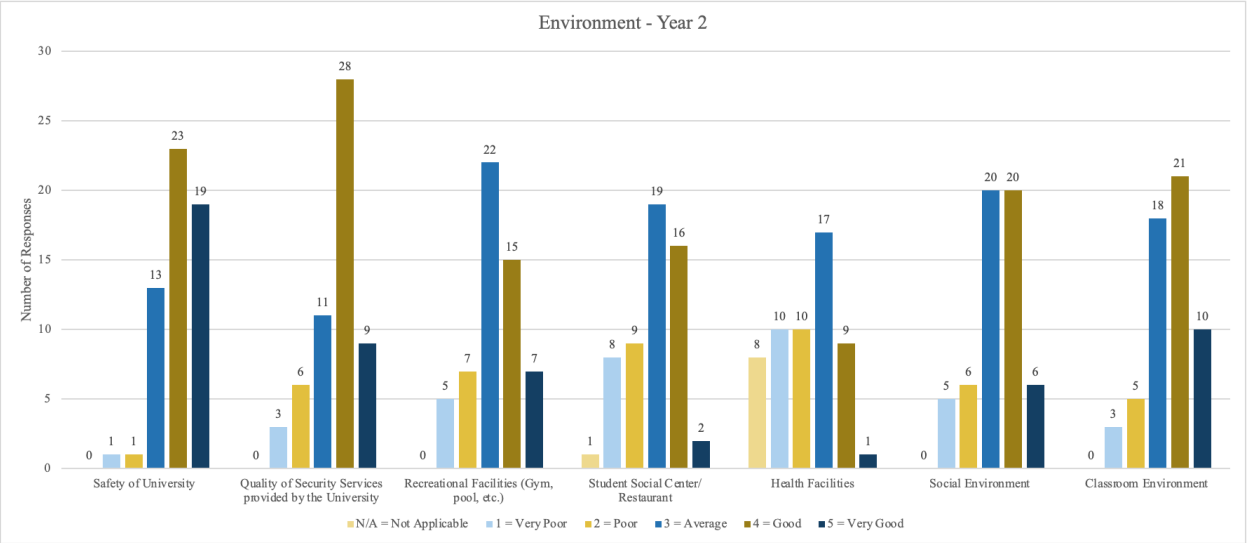
## Campus Environments



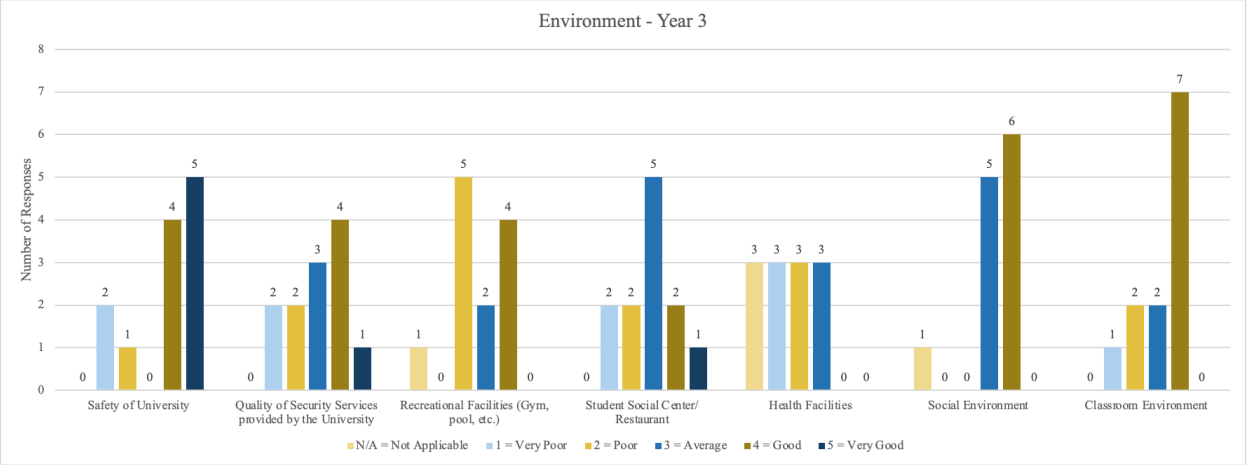

Overall Student Ratings of Various Environments and Spaces



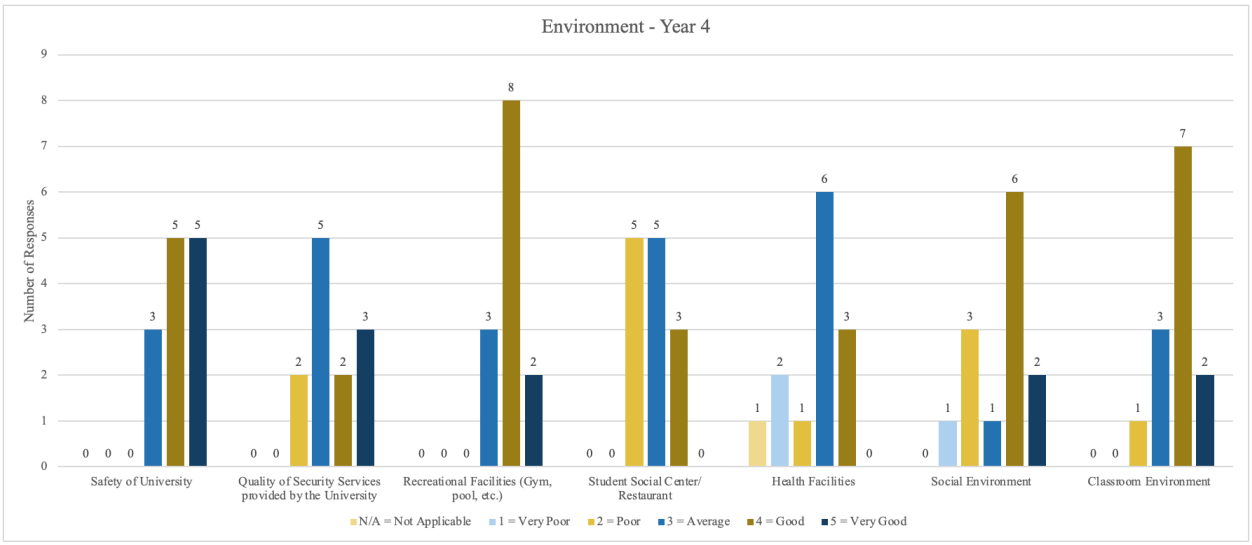
Year 1 Student Ratings of Various Environments and Spaces



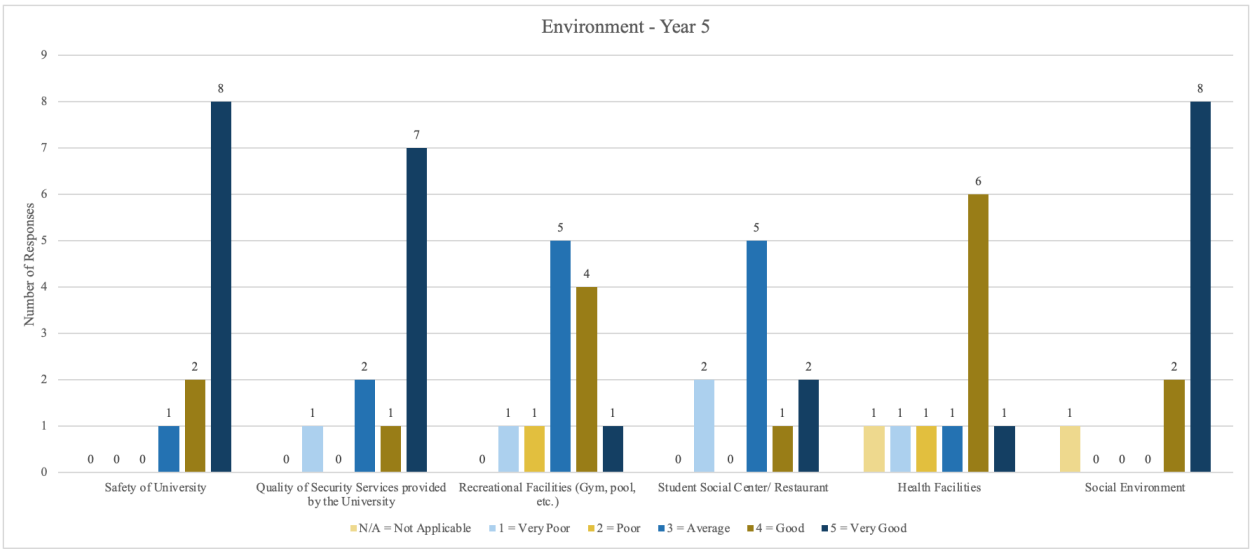

Year 2 Student Ratings of Various Environments and Spaces




Year 3 Student Ratings of Various Environments and Spaces

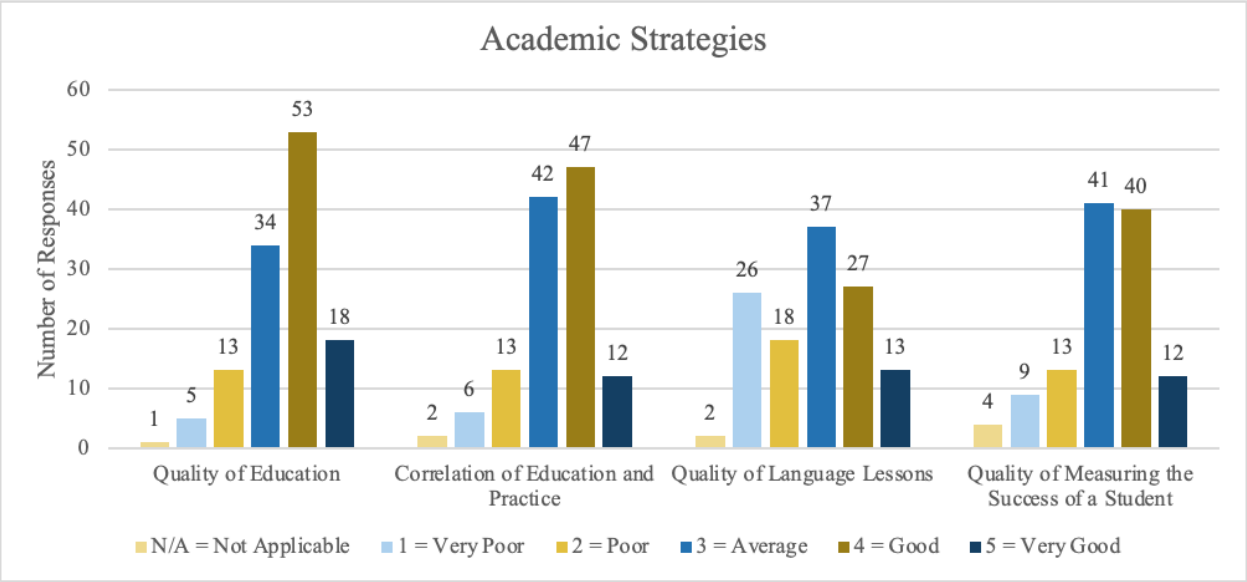



Year 4 Student Ratings of Various Environments and Spaces

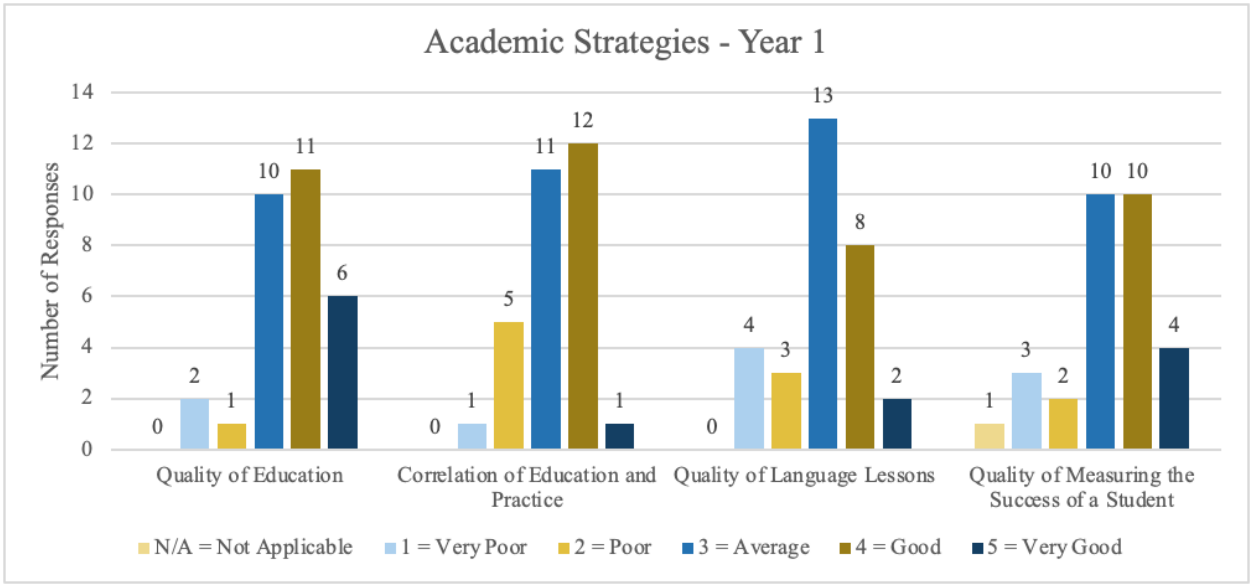



Year 5 Student Ratings of Various Environments and Spaces

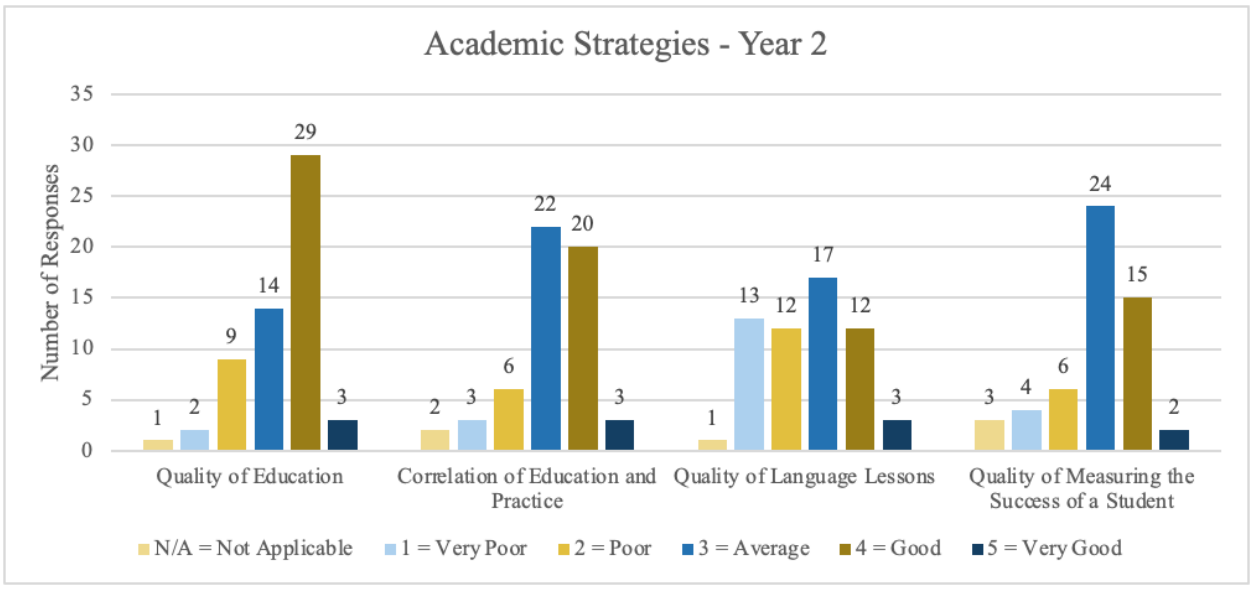
# Academic Strategies



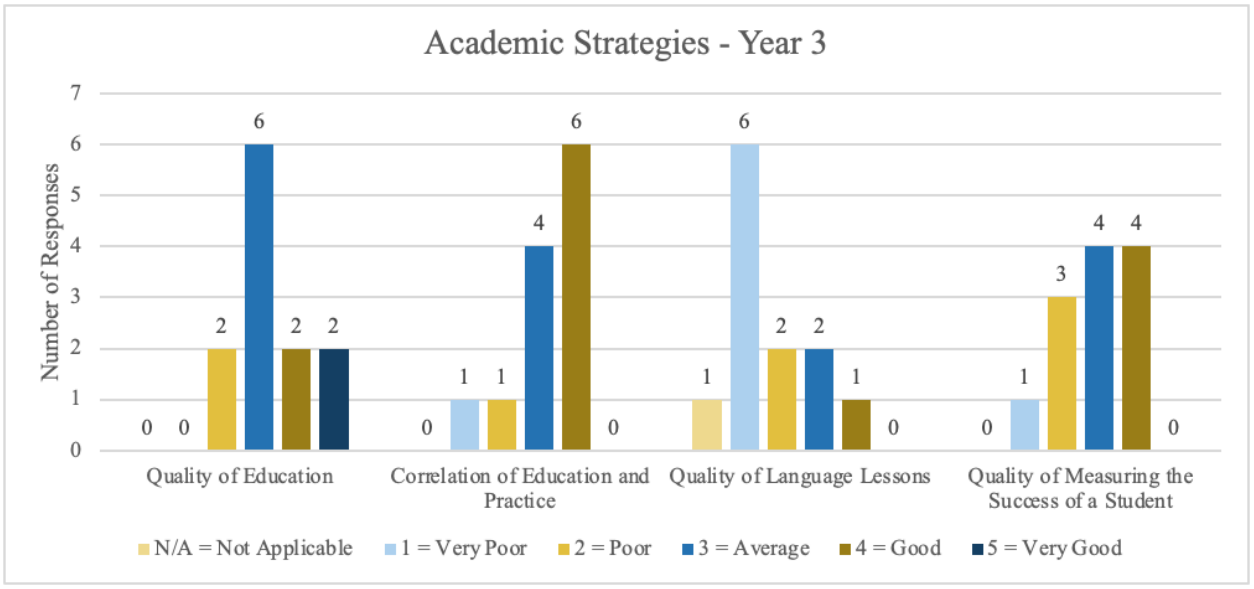

Overall Student Rankings of Academic Strategies



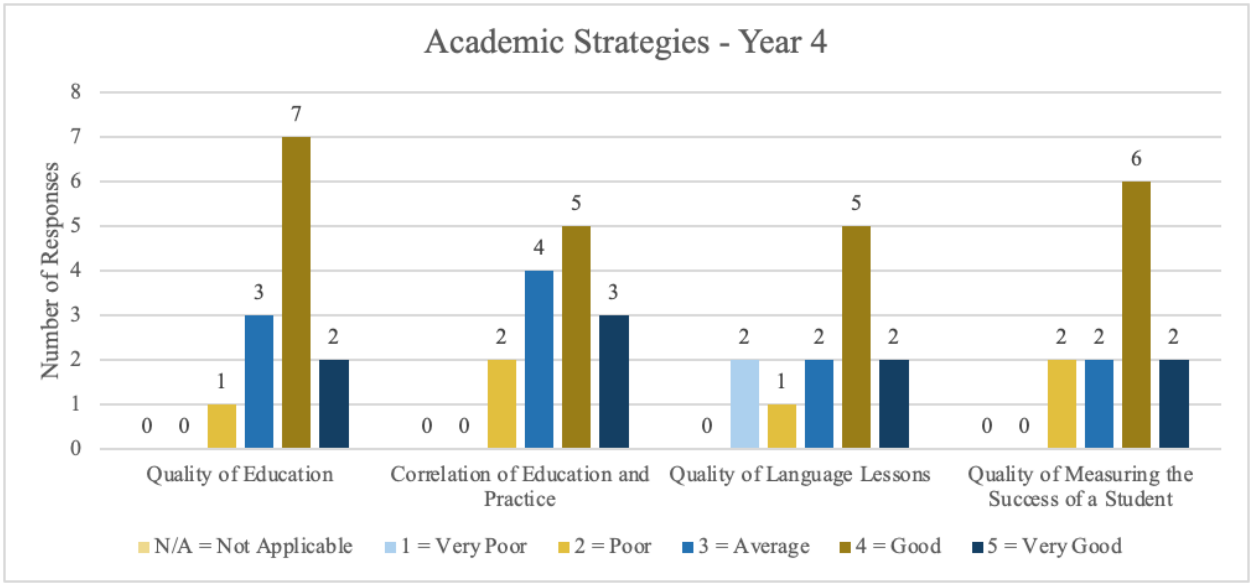

Year 1 Student Rankings of Academic Strategies



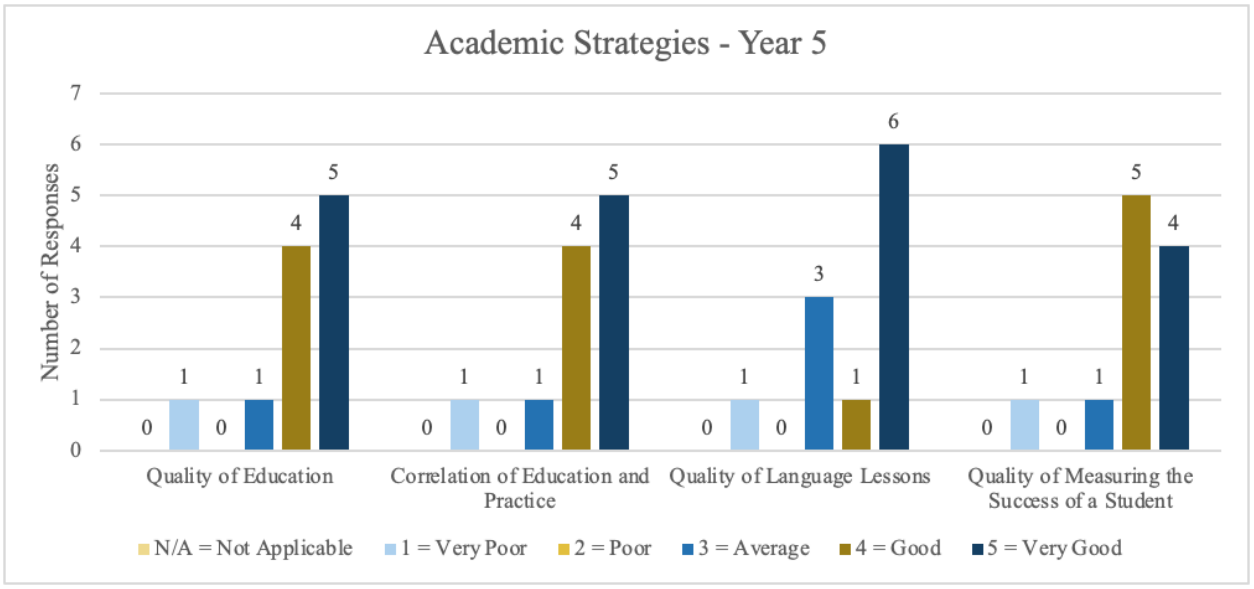

Year 2 Student Rankings of Academic Strategies




Year 3 Student Rankings of Academic Strategies

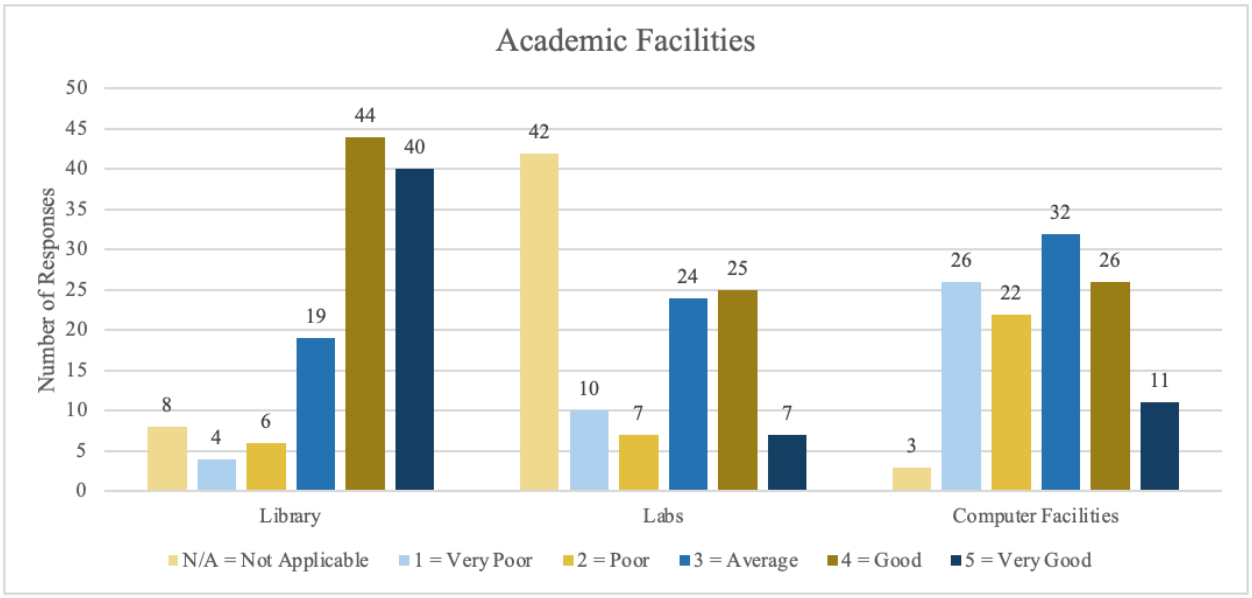



Year 4 Student Rankings of Academic Strategies

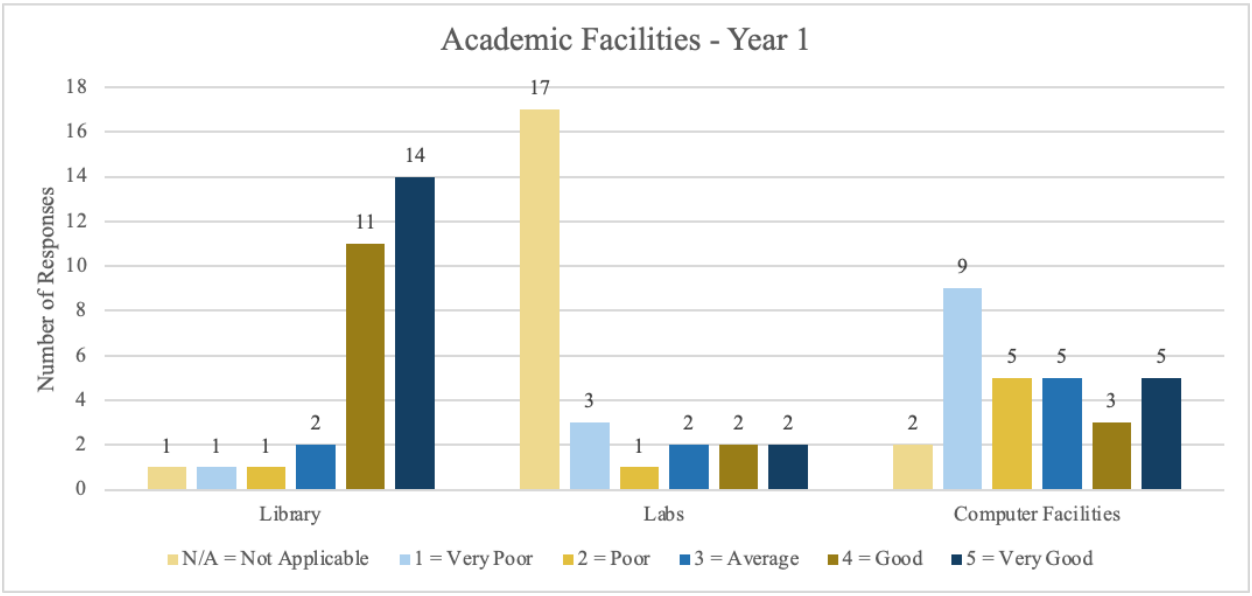



Year 5 Student Rankings of Academic Strategies

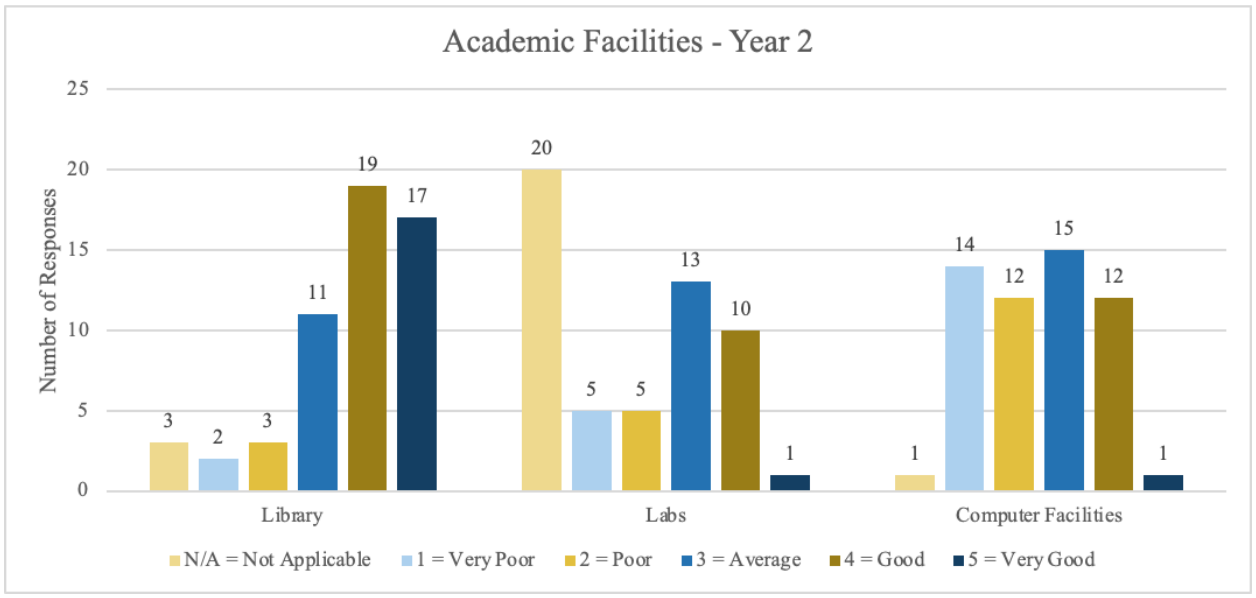
# Academic Facilities



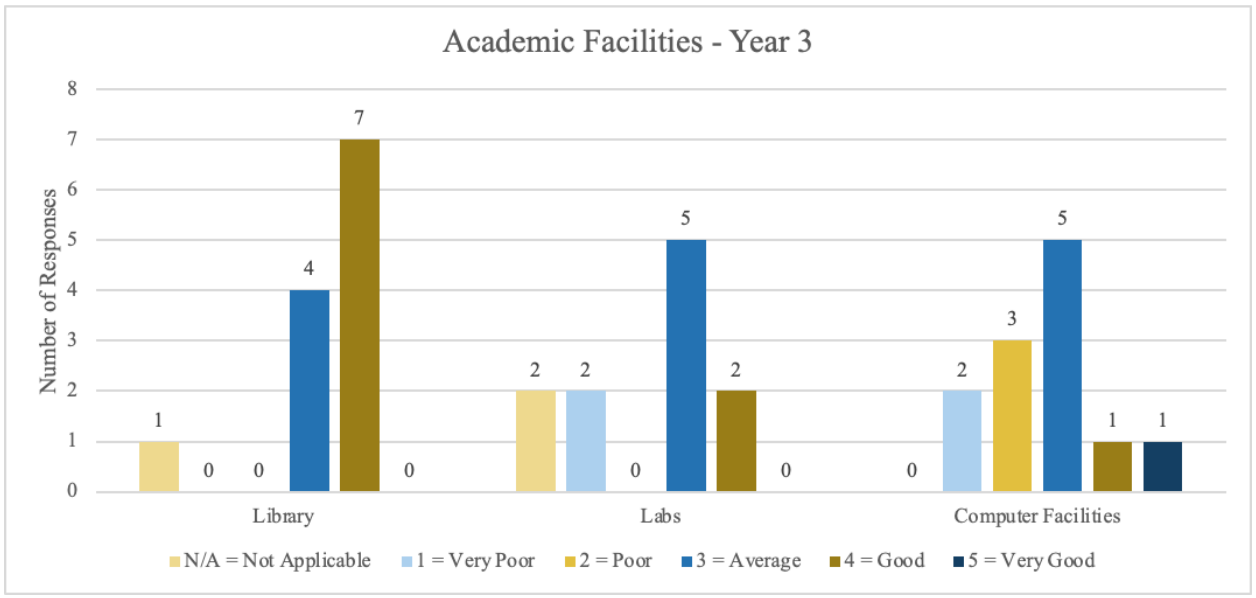

Overall Student Rankings of Academic Facilities



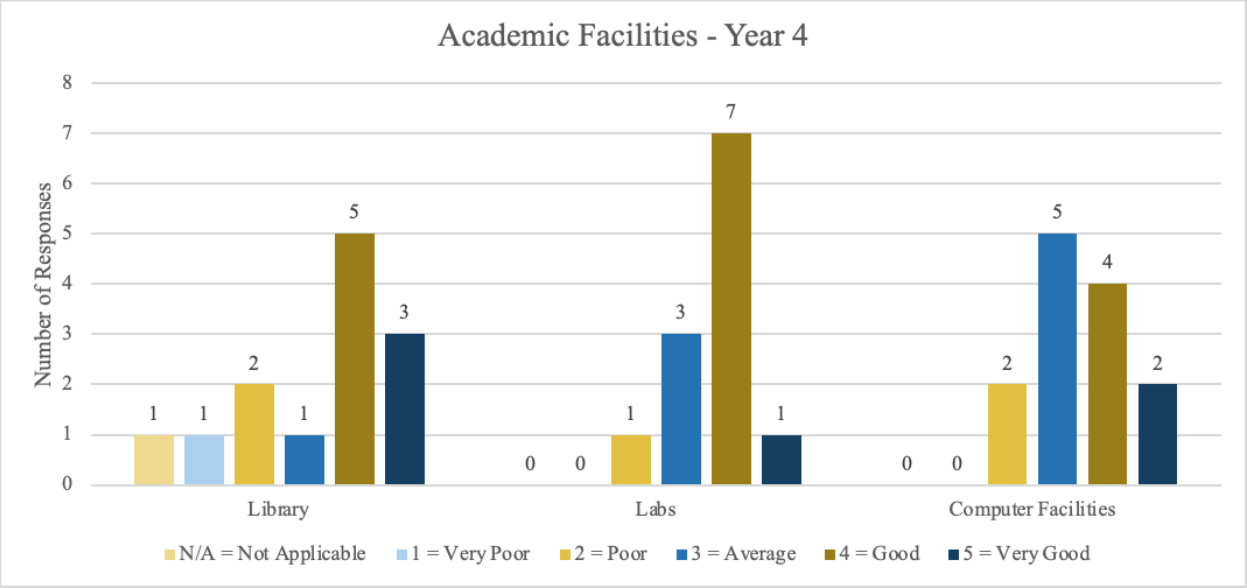

Year 1 Student Rankings of Academic Facilities



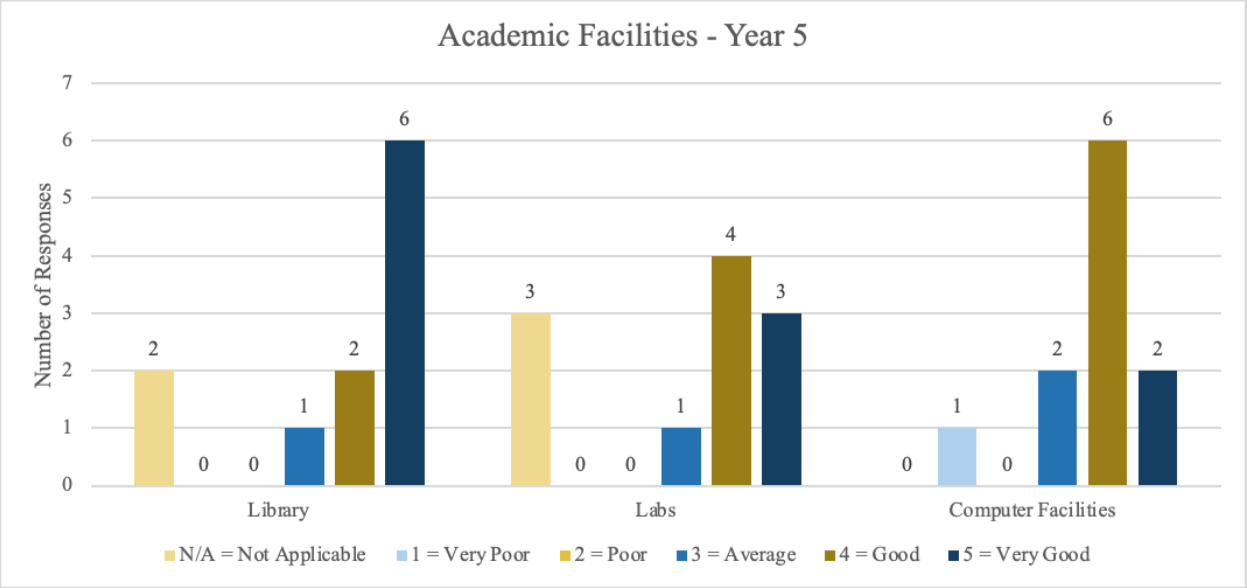

Year 2 Student Rankings of Academic Facilities




Year 3 Student Rankings of Academic Facilities

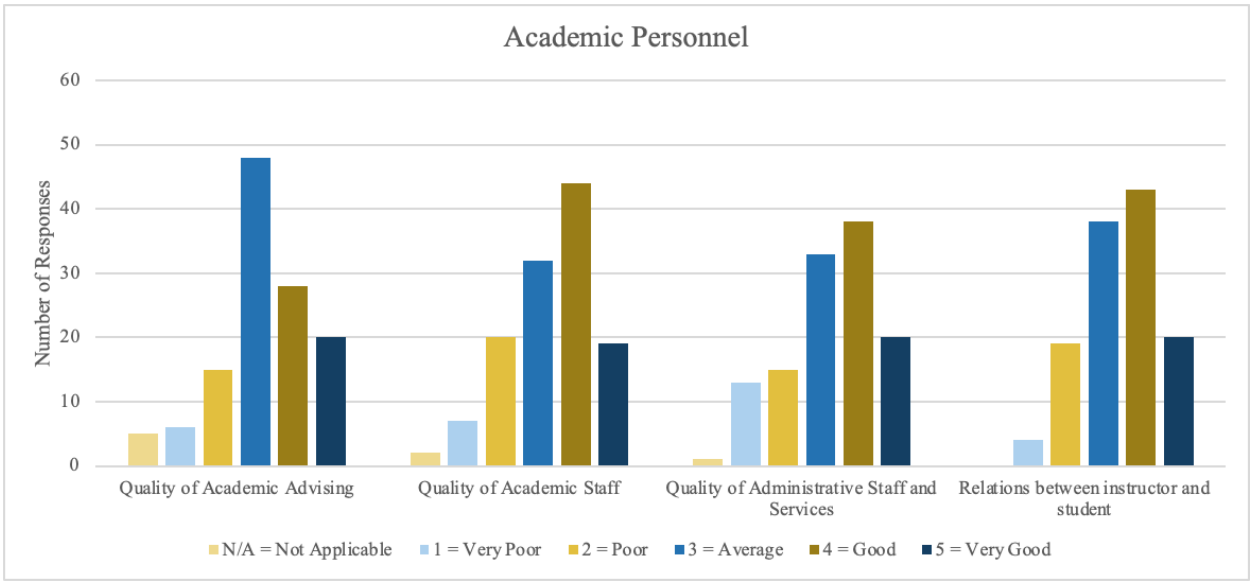



Year 4 Student Rankings of Academic Facilities

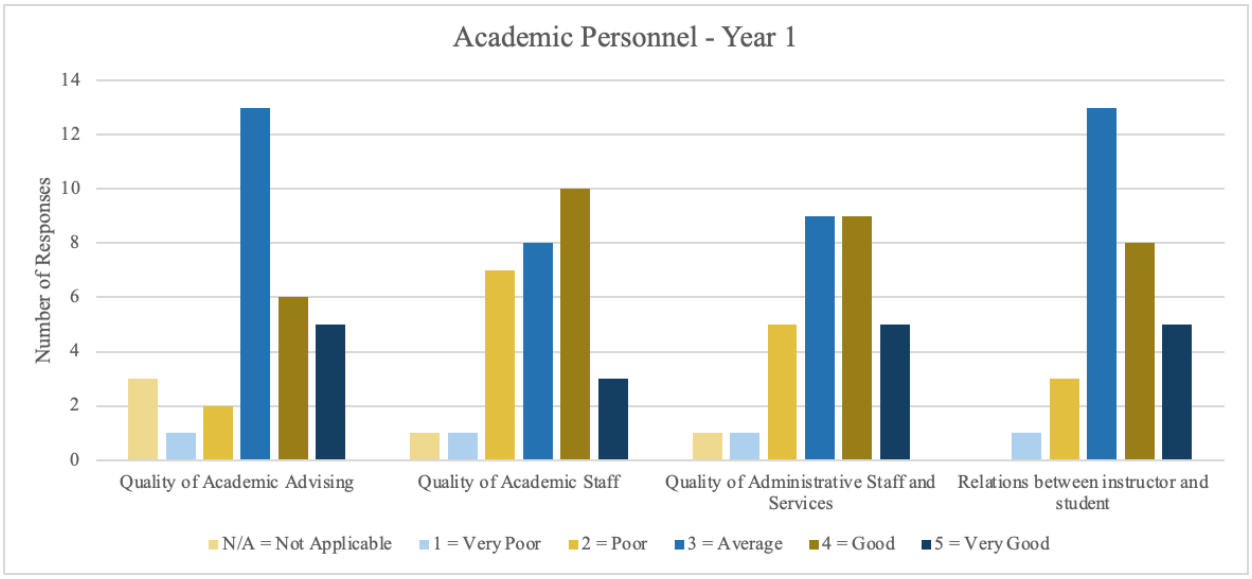



Year 5 Student Rankings of Academic Facilities

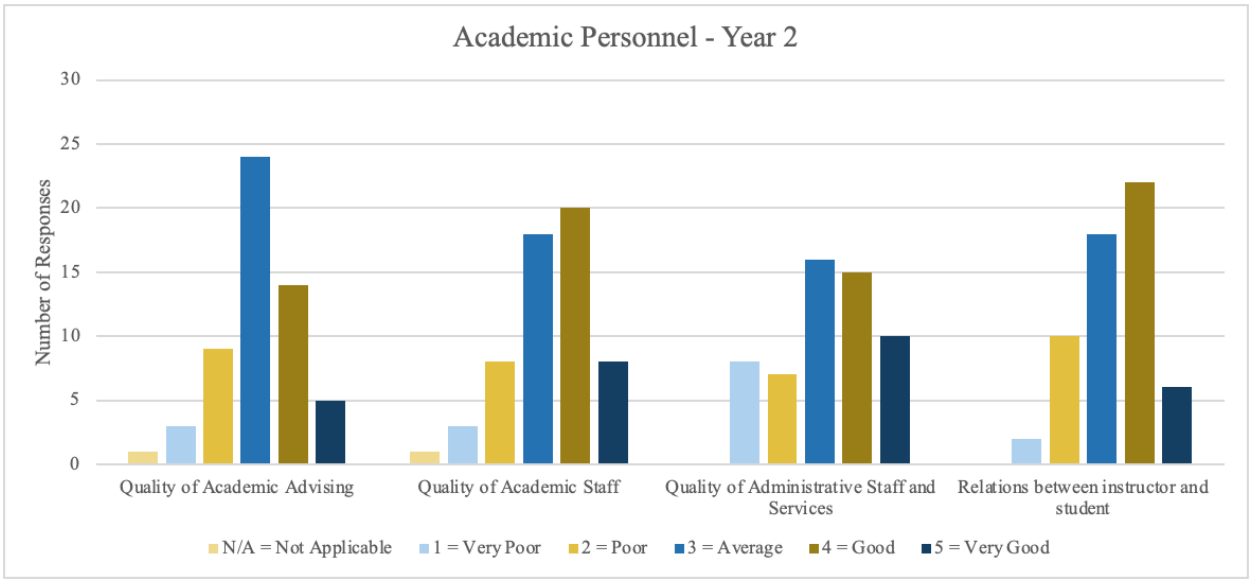
# Academic Personnel



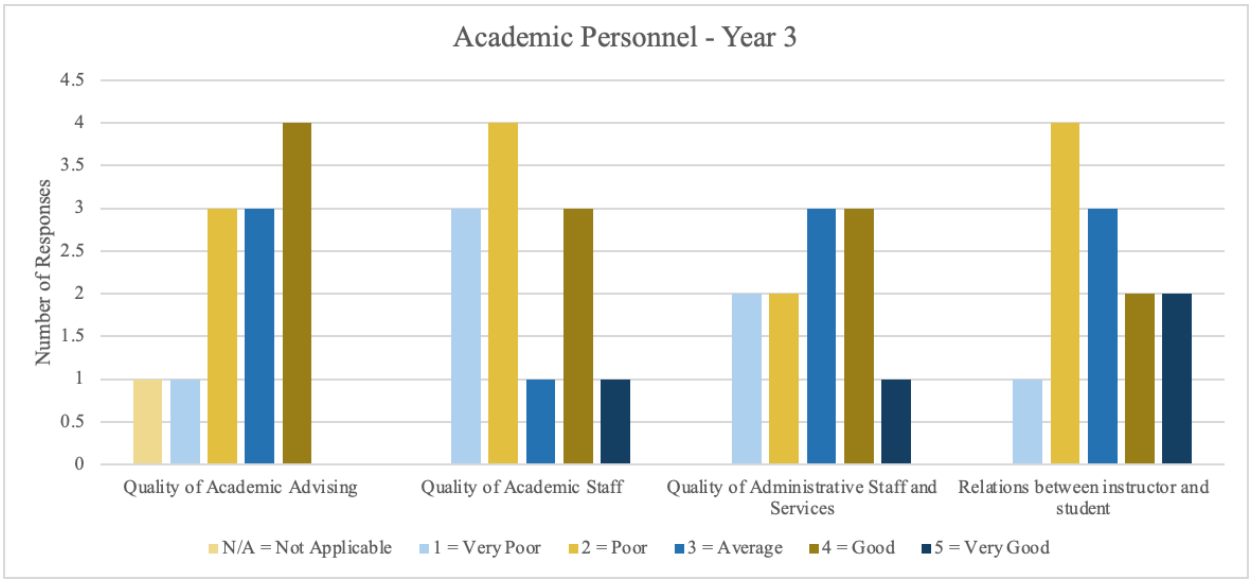

Overall Student Rating of Academic Personnel



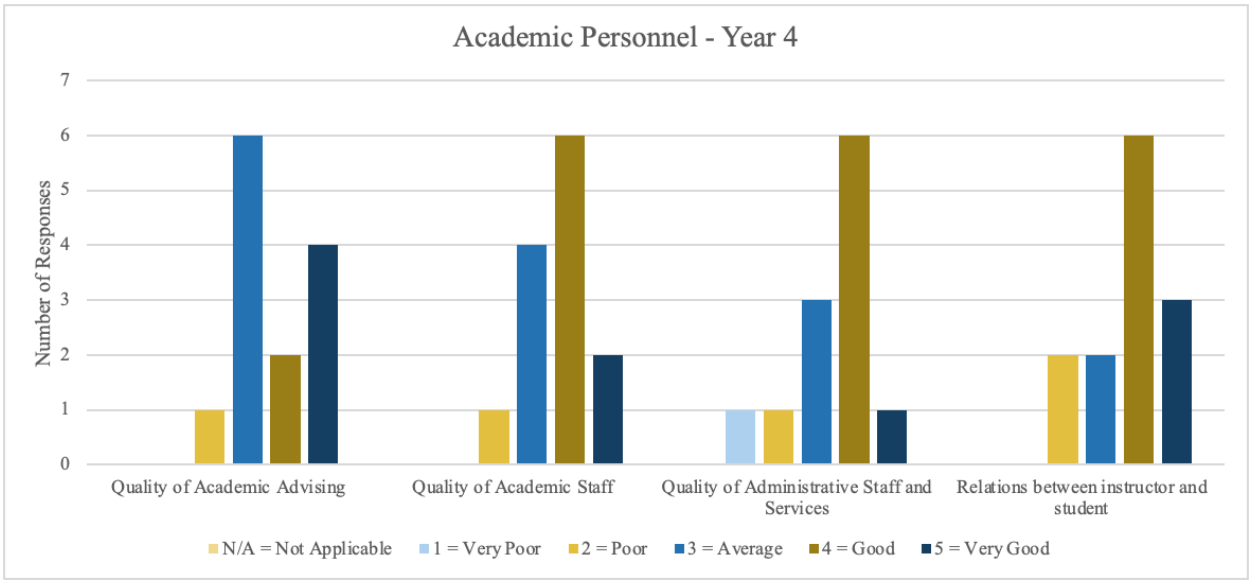

Year 1 Student Rating of Academic Personnel



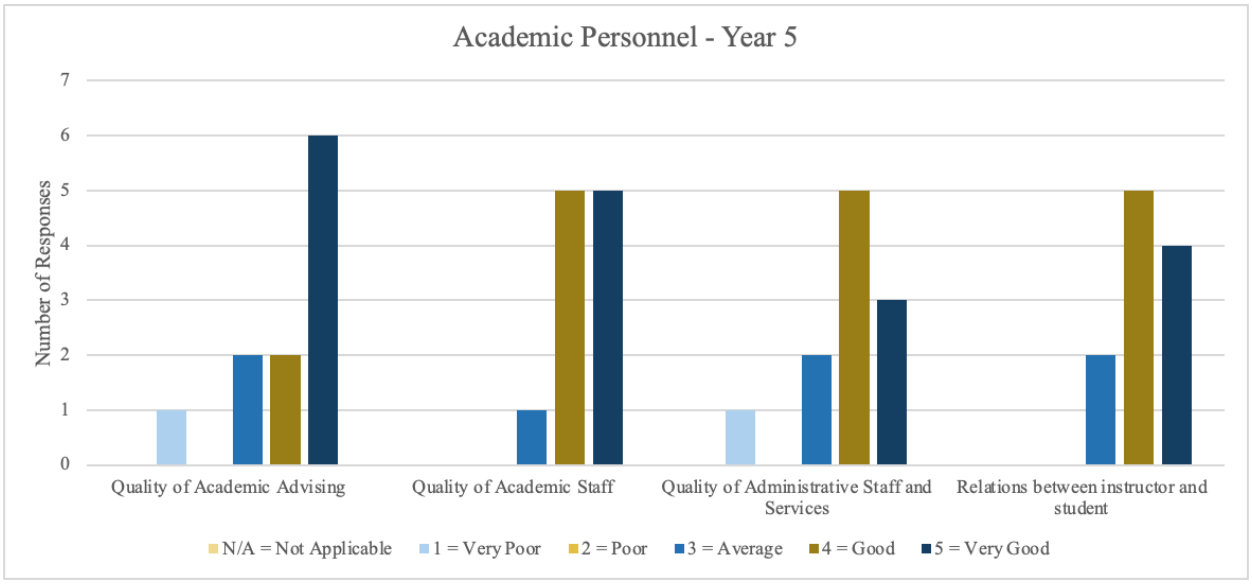

Year 2 Student Rating of Academic Personnel




Year 3 Student Rating of Academic Personnel

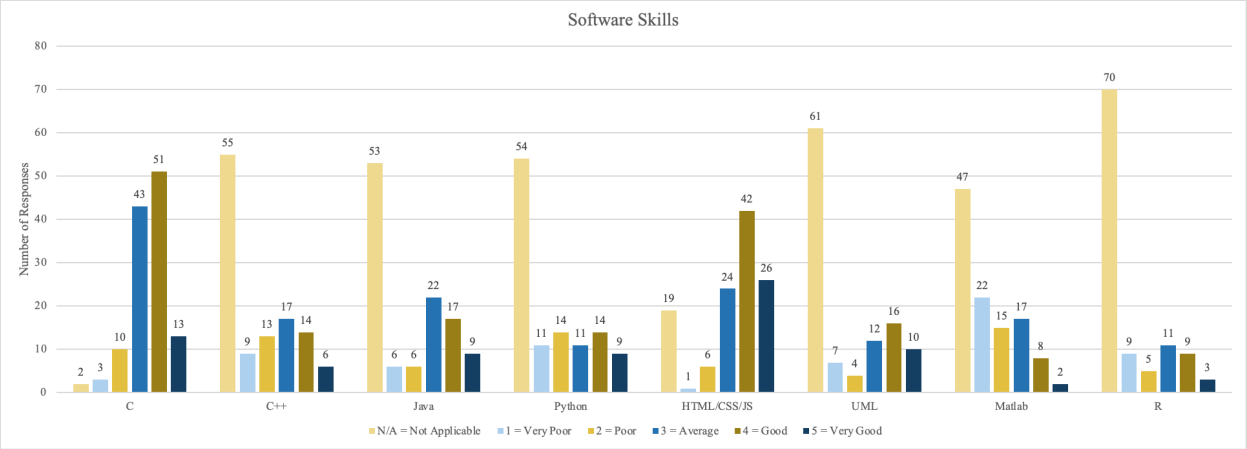



Year 4 Student Rating of Academic Personnel

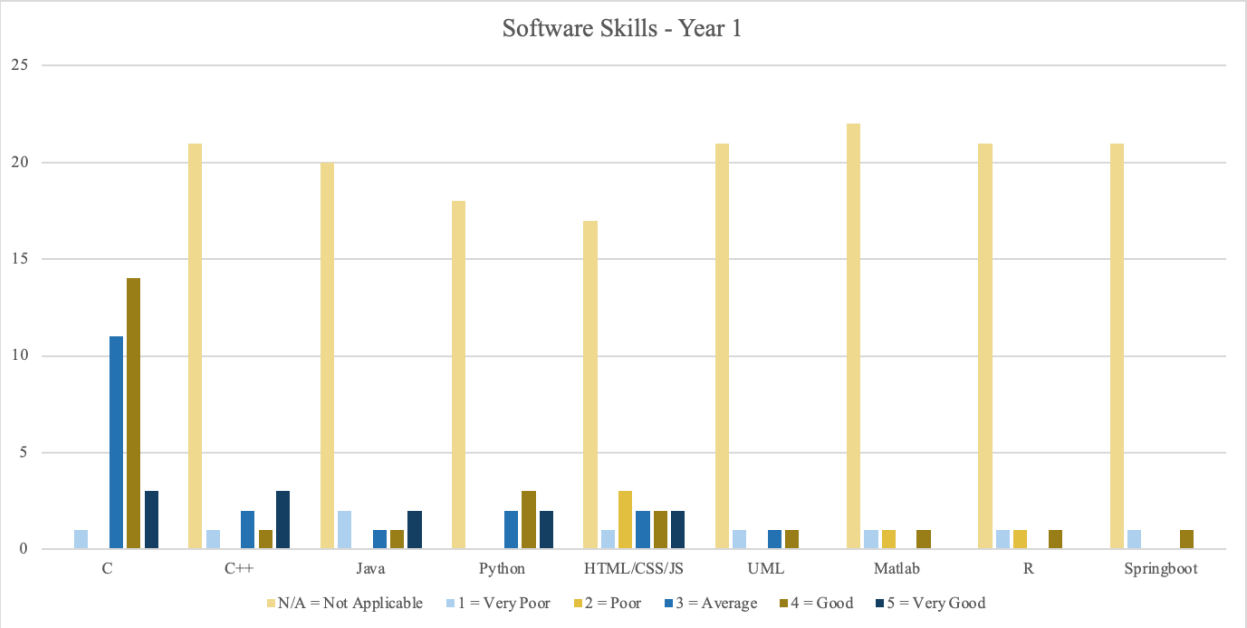



Year 5 Student Rating of Academic Personnel

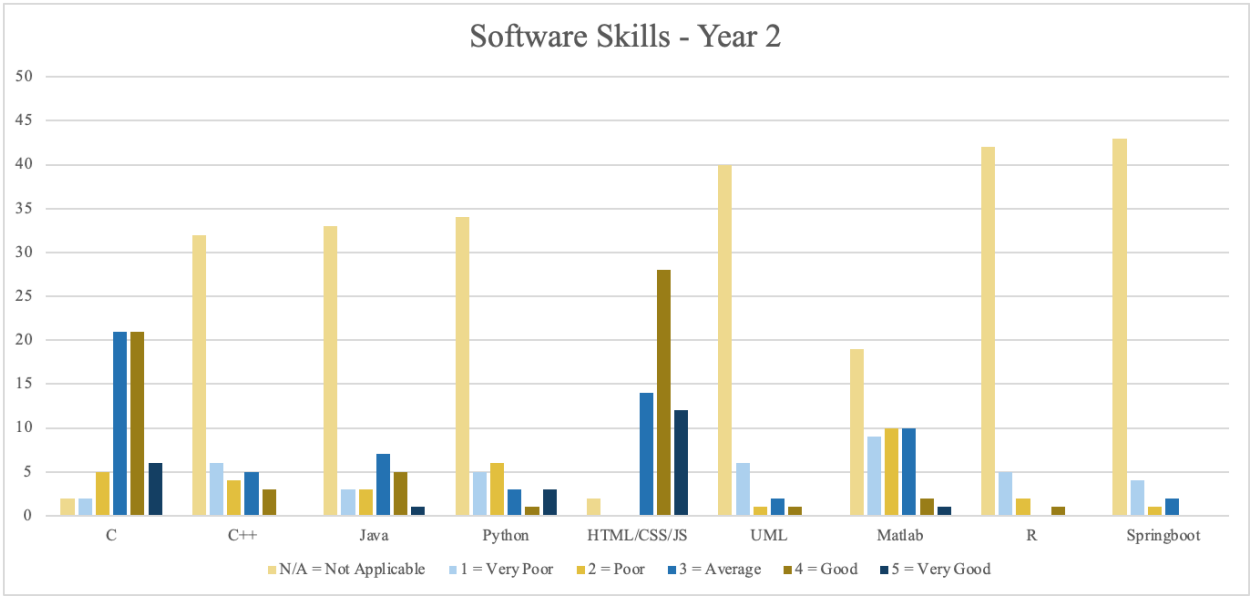
# Software Skills



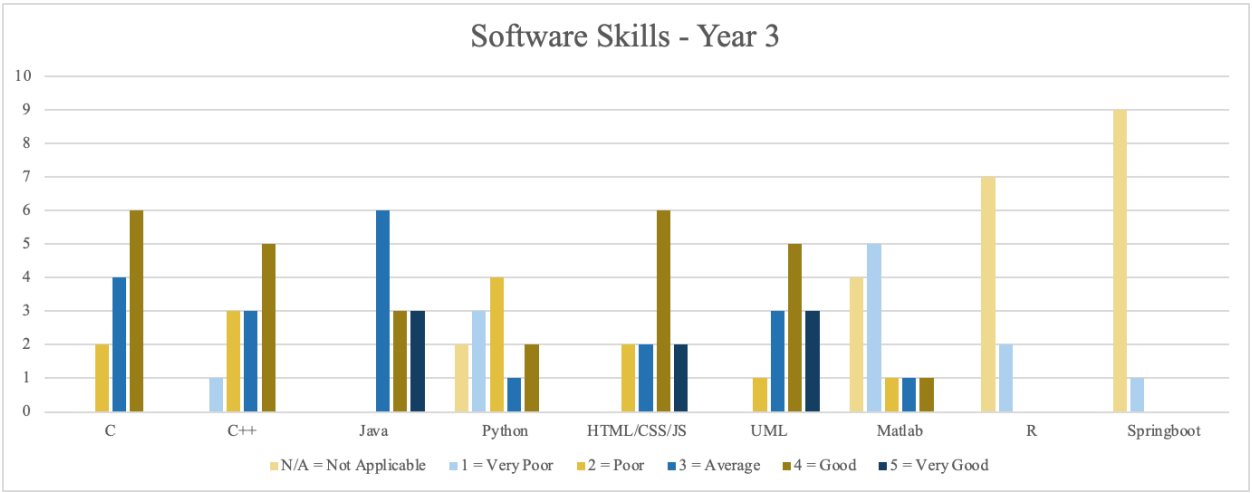

Overall Student Ranking of Software Skills



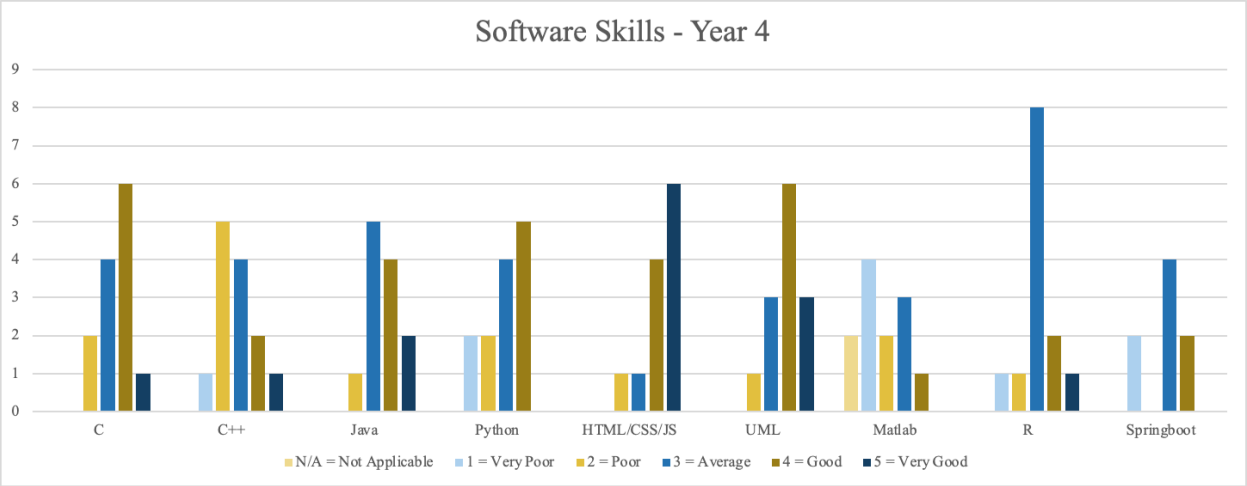

Year 1 Student Ranking of Software Skills



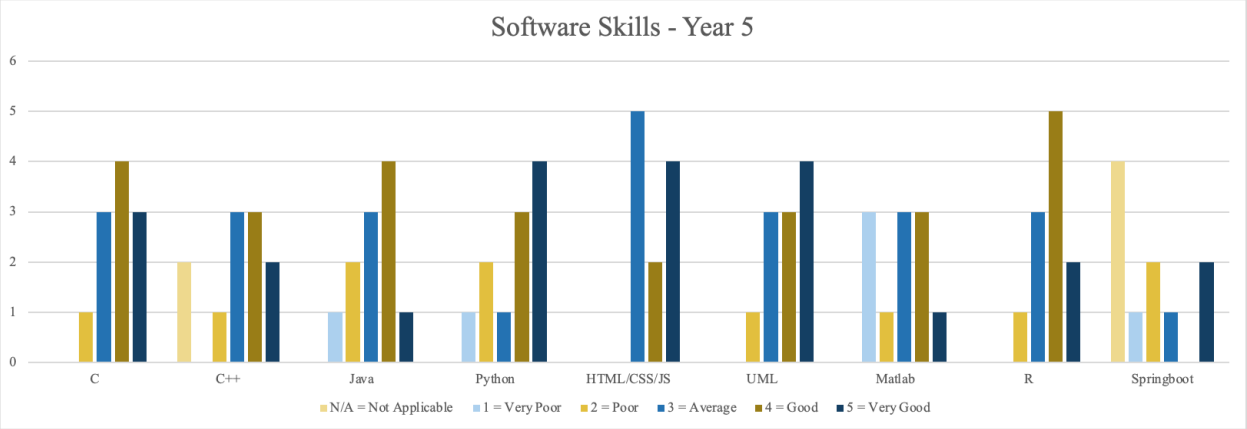

Year 2 Student Ranking of Software Skills




Year 3 Student Ranking of Software Skills

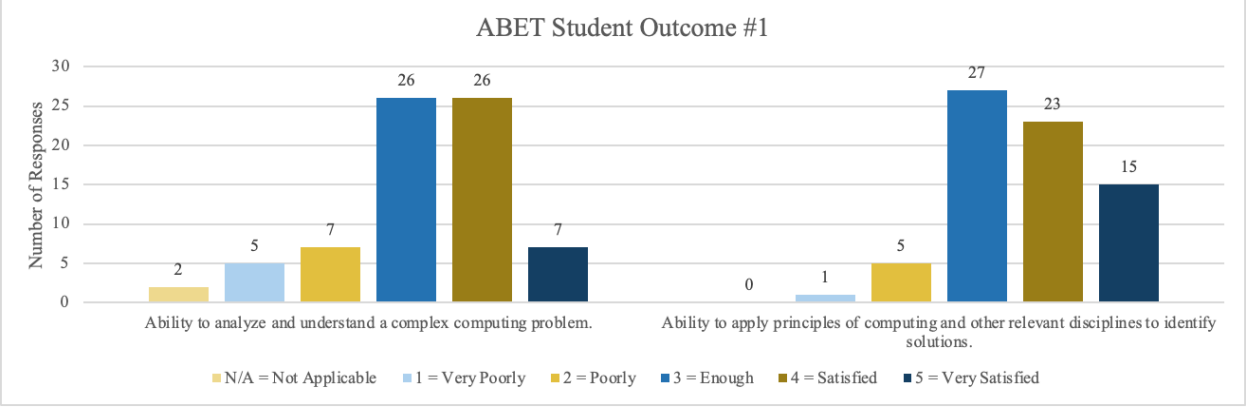



Year 4 Student Ranking of Software Skills

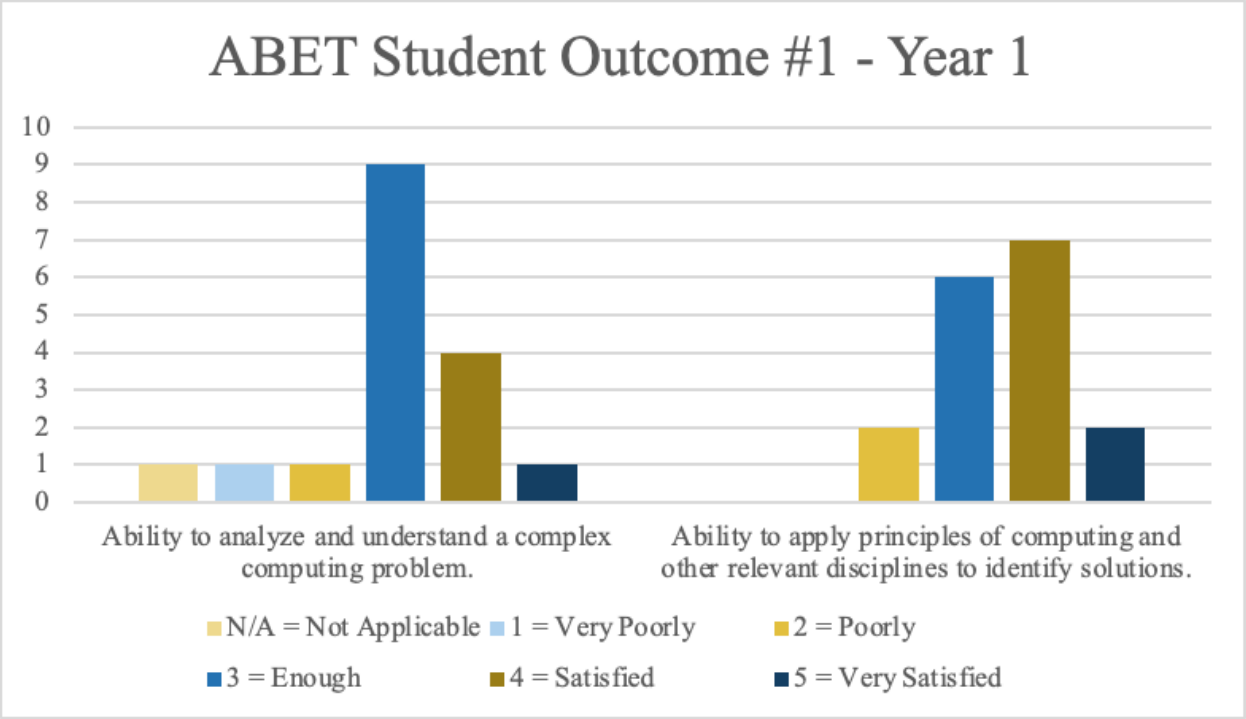



Year 5 Student Ranking of Software Skills

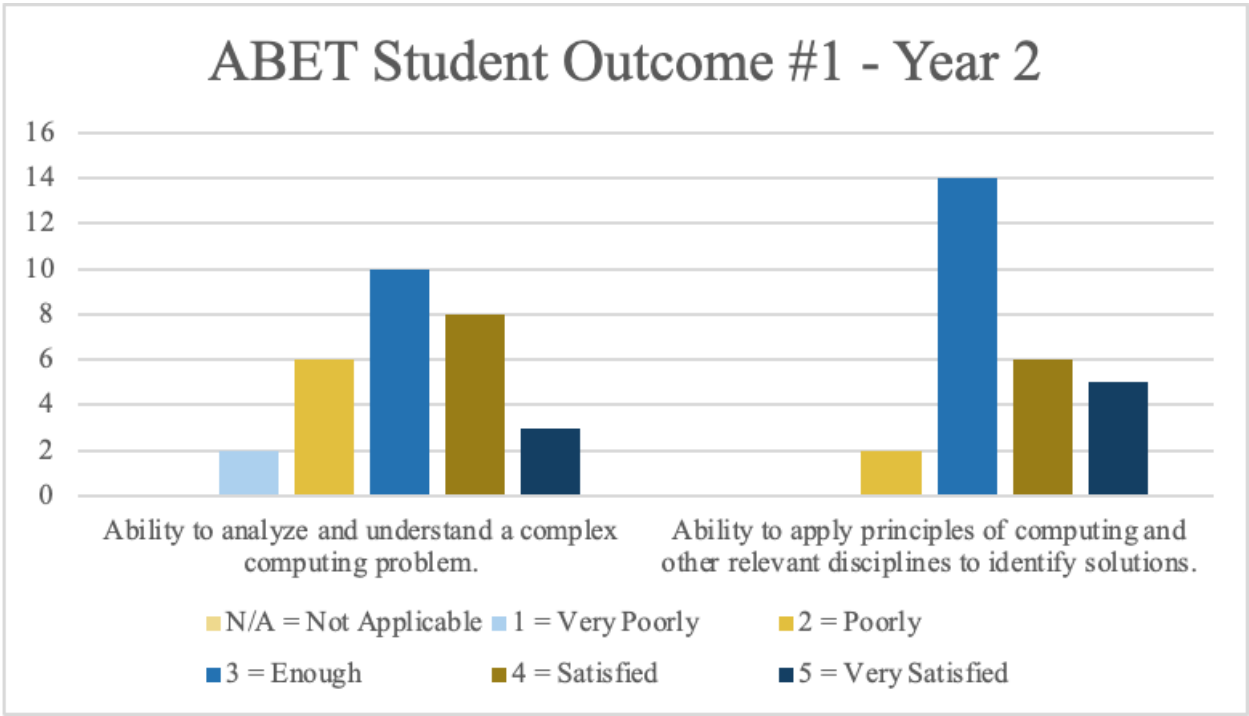
# Student Outcome #1



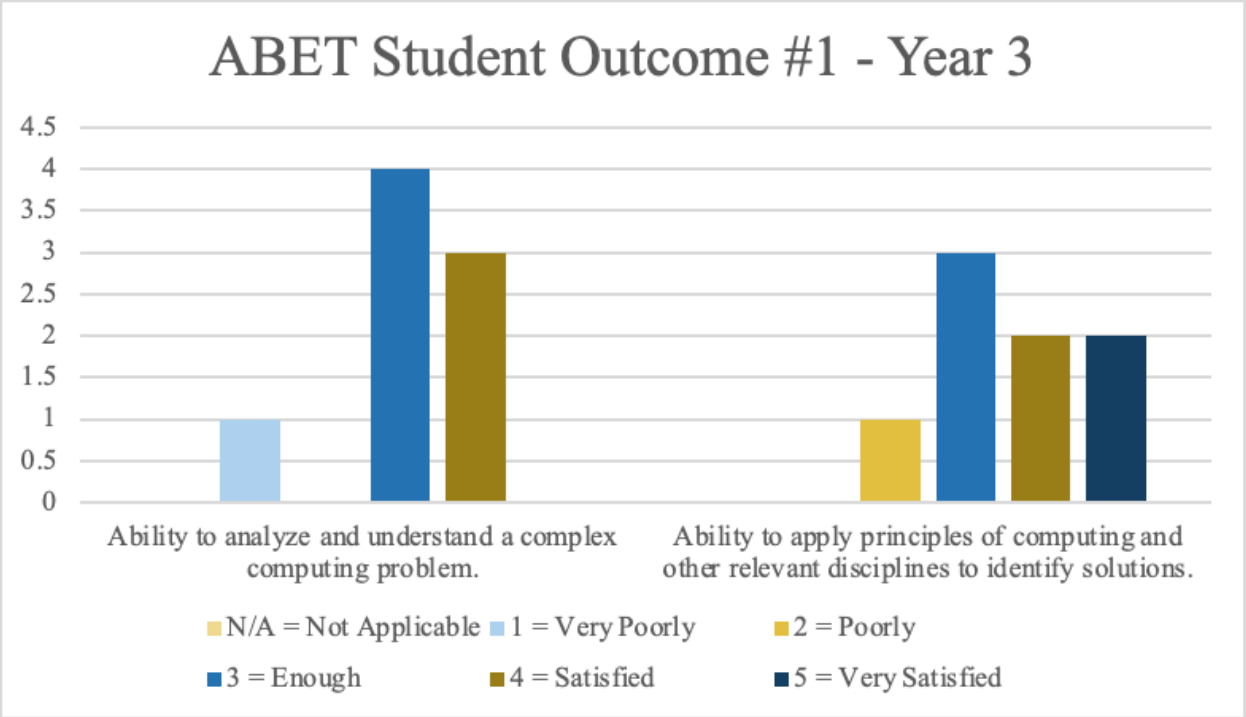
Overall Rating of Abilities in ABET Student Outcome 1



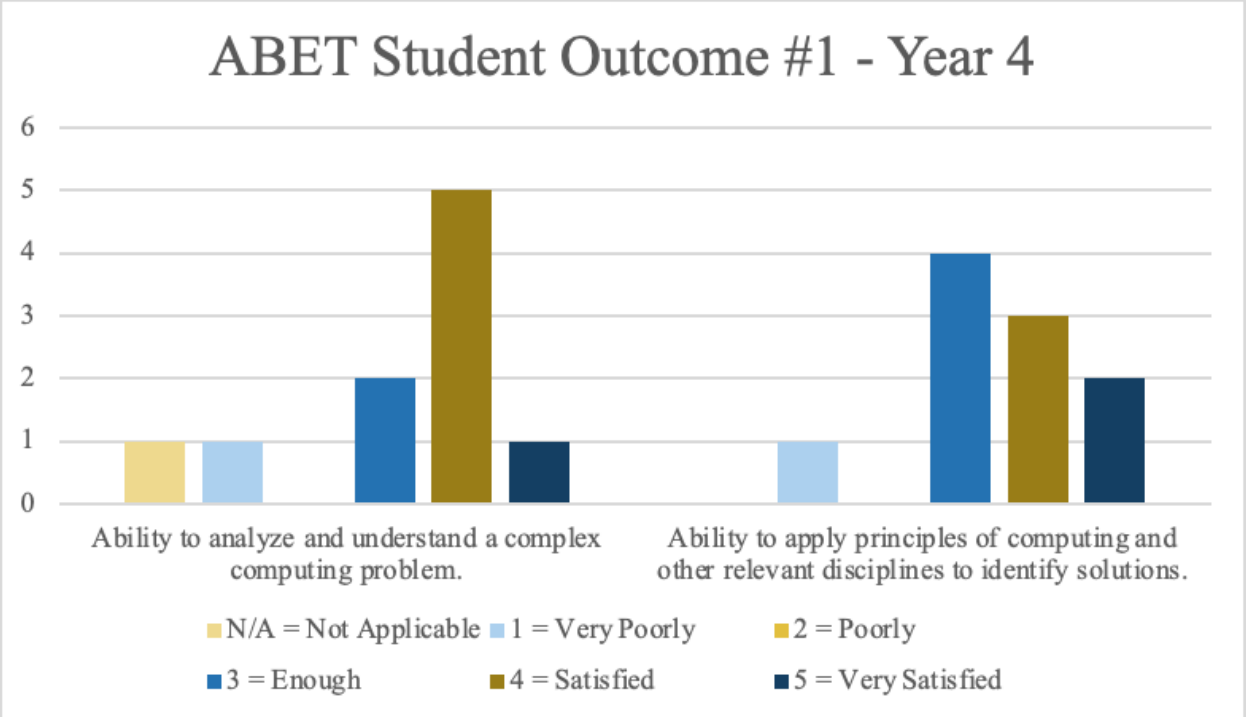

Year 1 Rating of Abilities in ABET Student Outcome 1



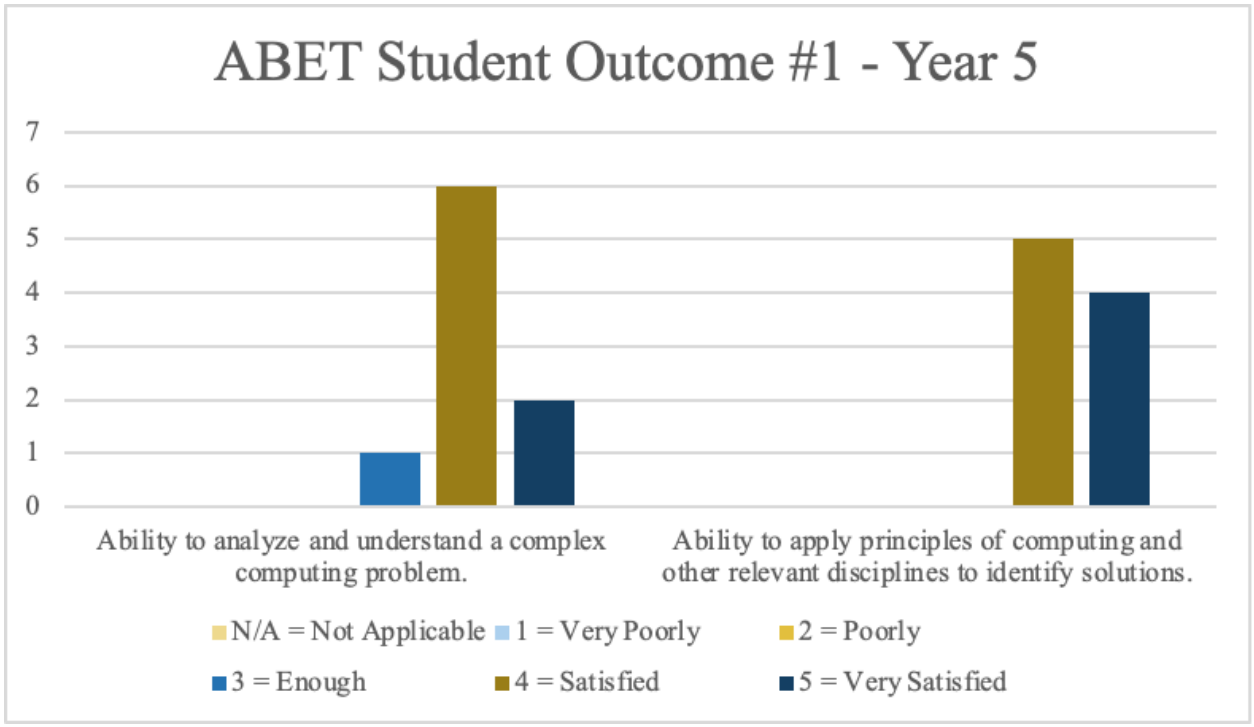
Year 2 Rating of Abilities in ABET Student Outcome 1




Year 3 Rating of Abilities in ABET Student Outcome 1

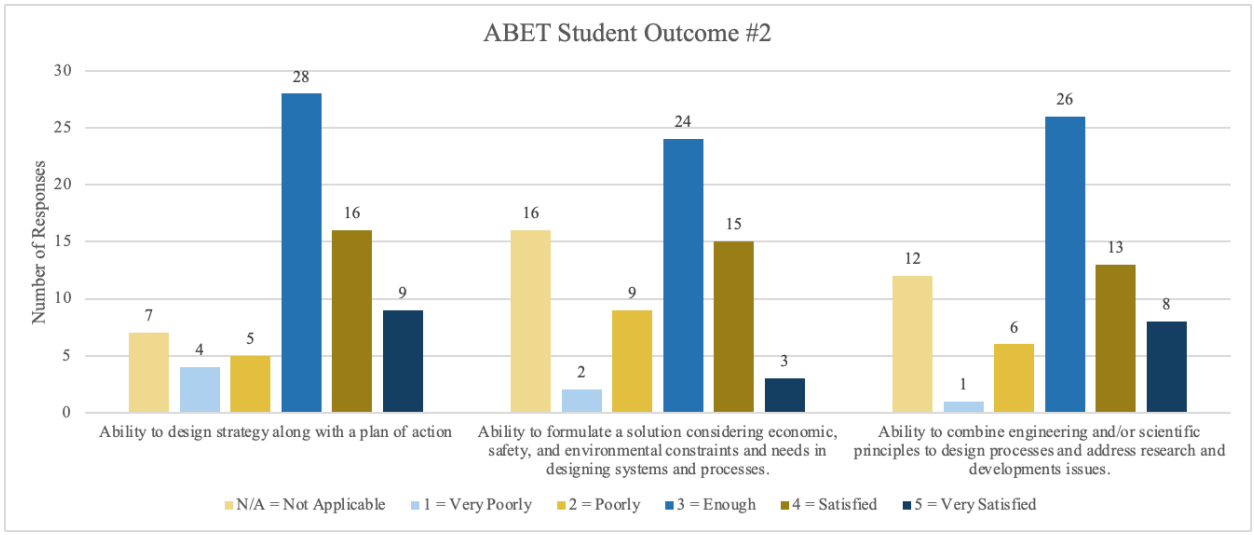



Year 4 Rating of Abilities in ABET Student Outcome 1

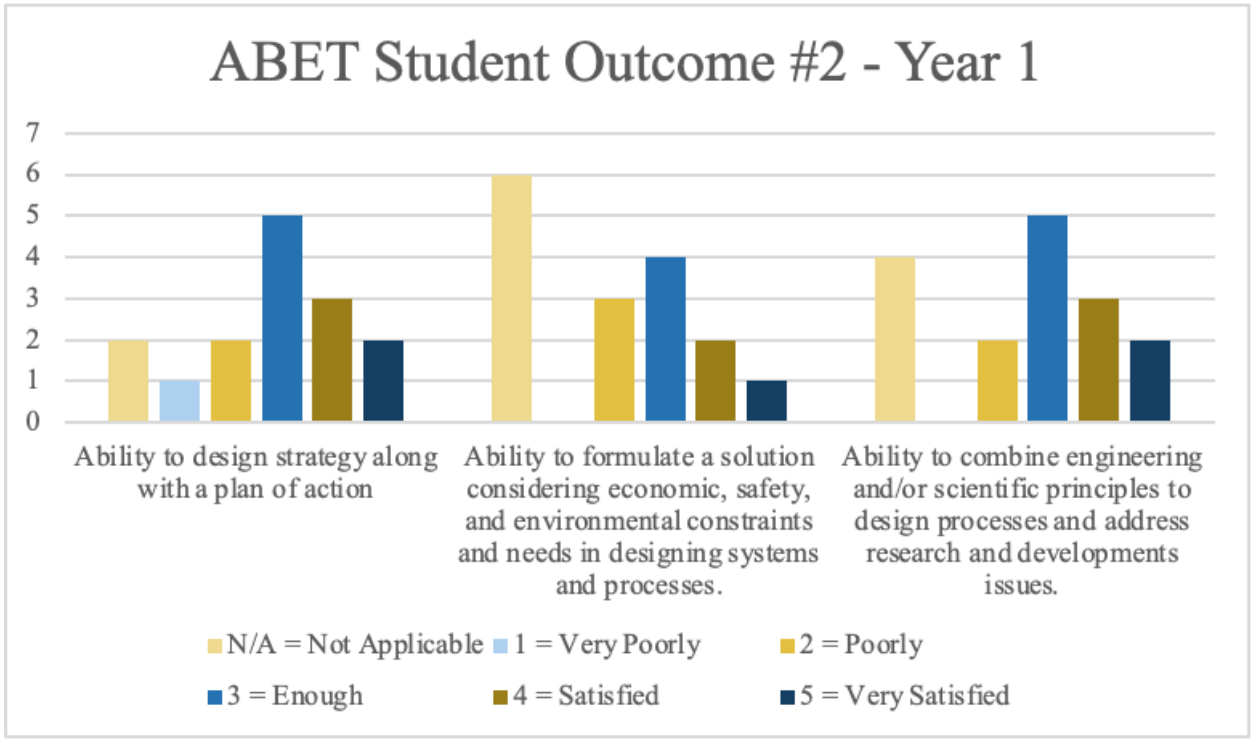



Year 5 Rating of Abilities in ABET Student Outcome 1

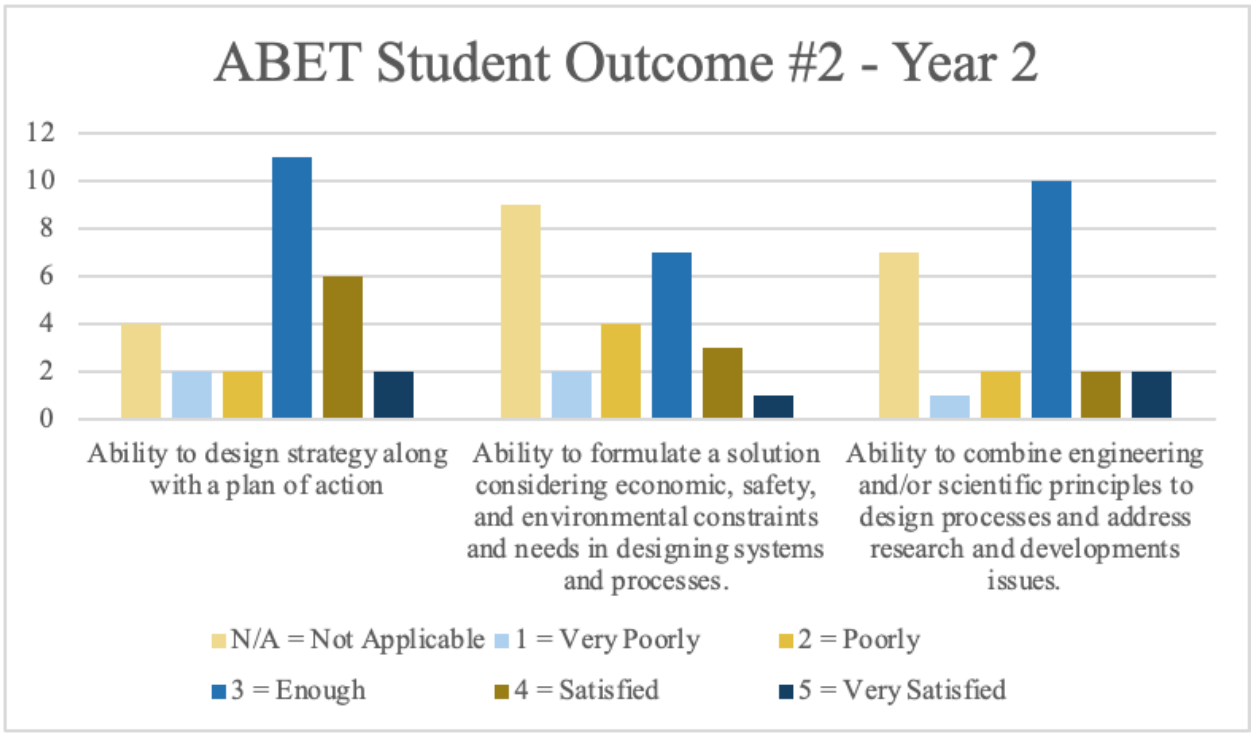
## Student Outcome #2



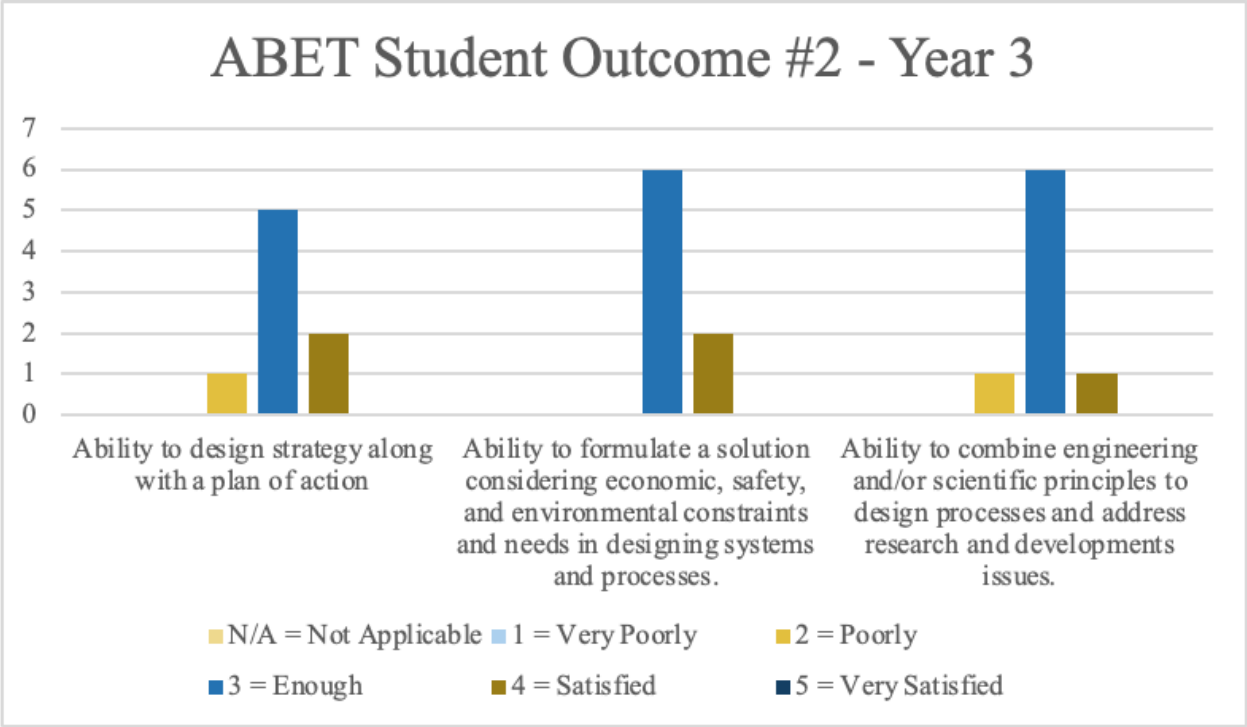

Overall Rating of Abilities in ABET Student Outcome 2



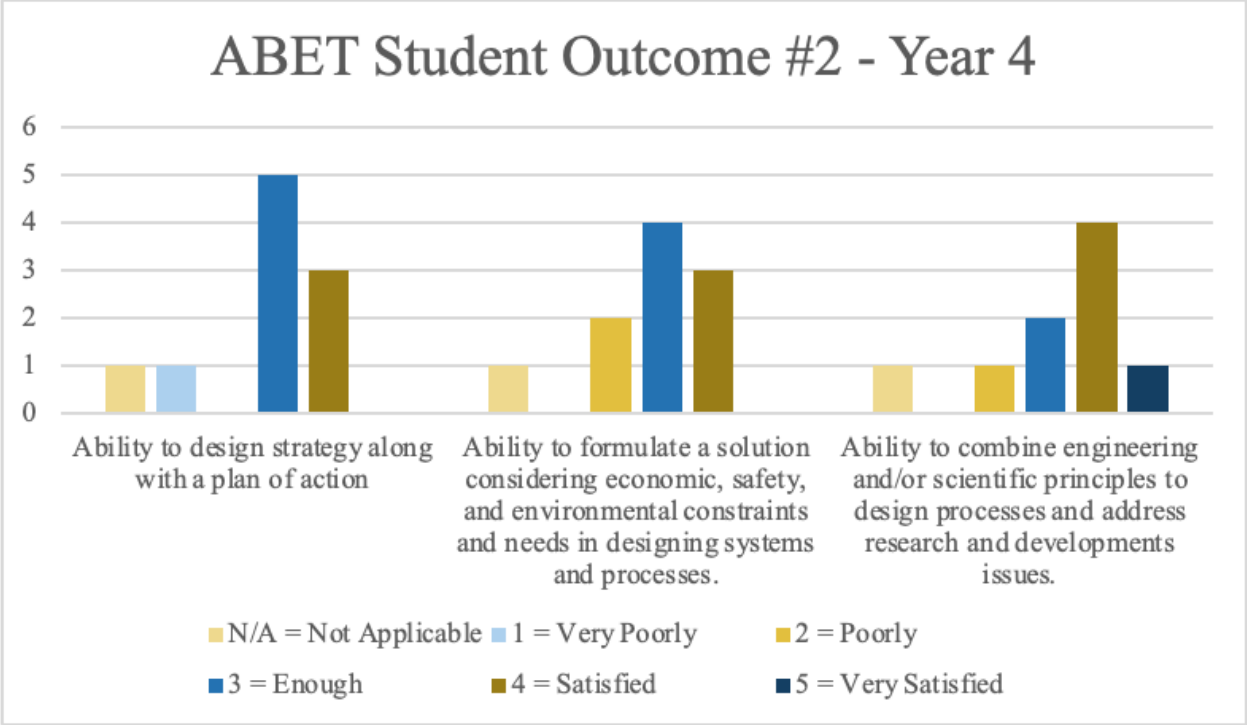

Year 1 Rating of Abilities in ABET Student Outcome 2



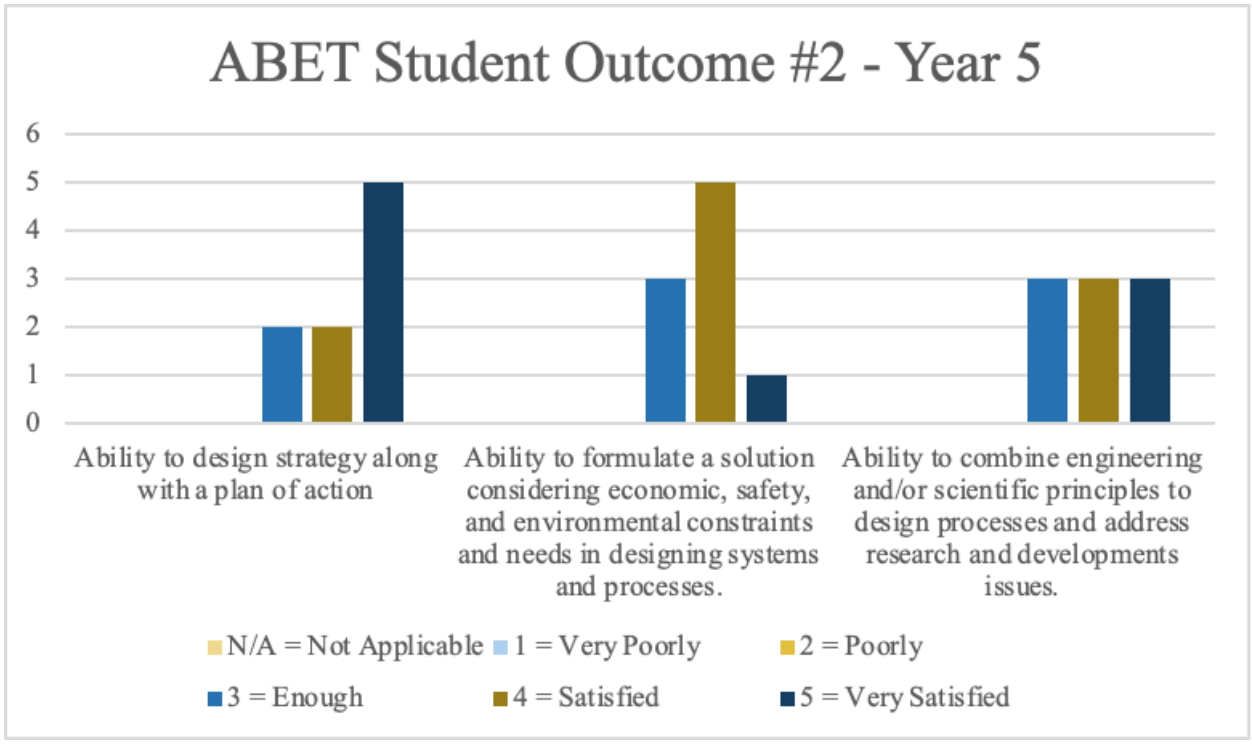

Year 2 Rating of Abilities in ABET Student Outcome 2




Year 3 Rating of Abilities in ABET Student Outcome 2

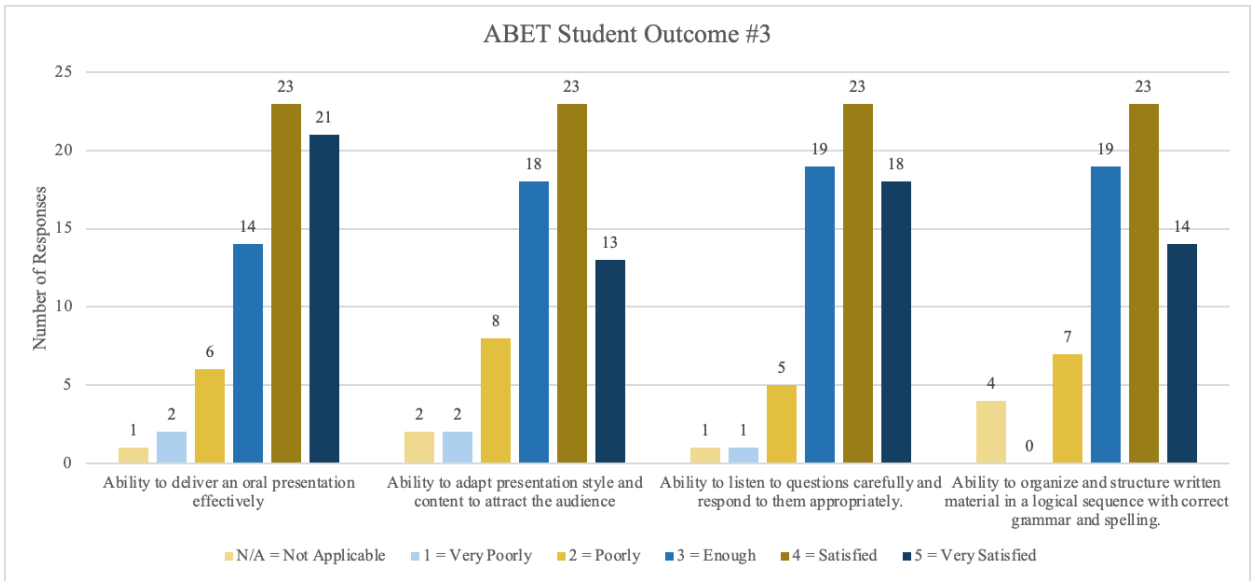



Year 4 Rating of Abilities in ABET Student Outcome 2

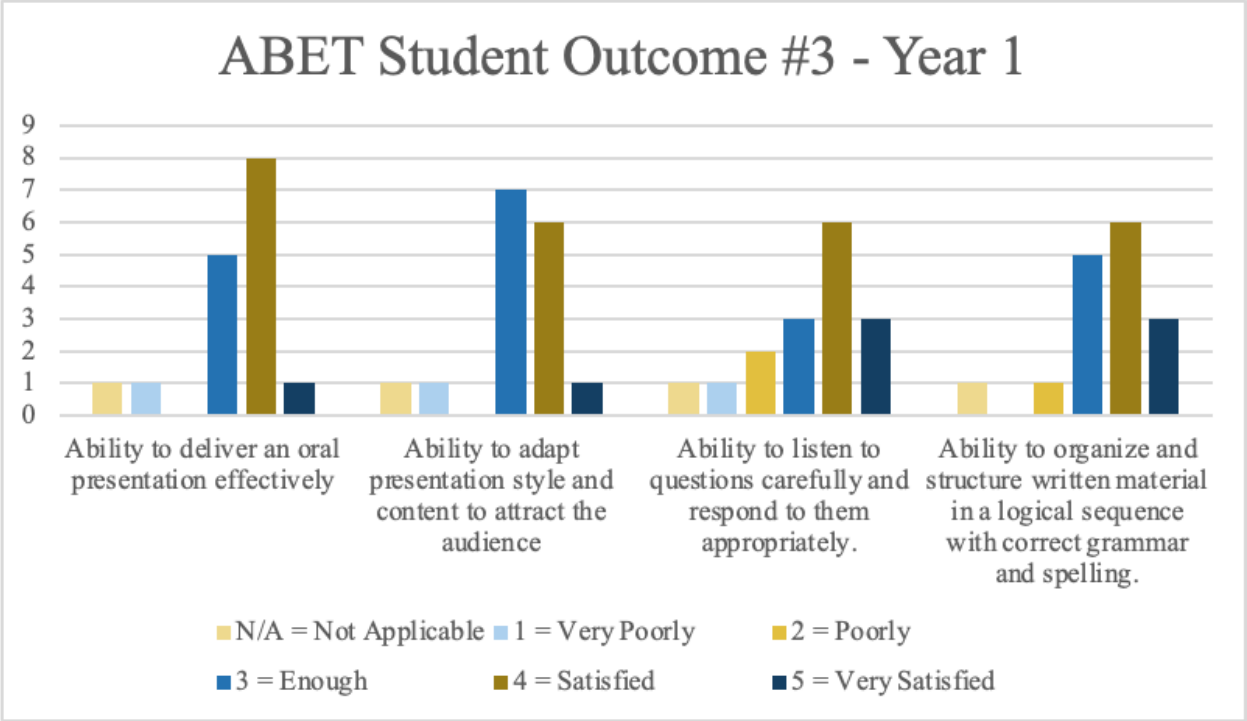



Year 5 Rating of Abilities in ABET Student Outcome 2

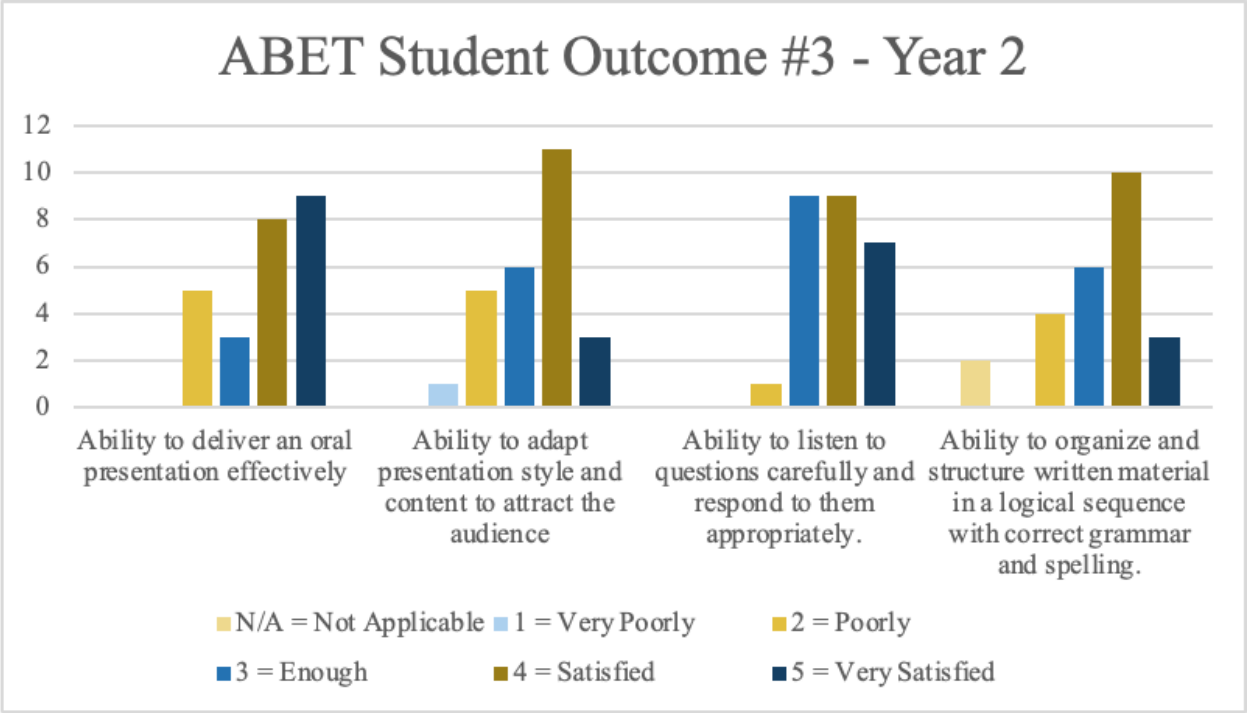
## Student Outcome #3



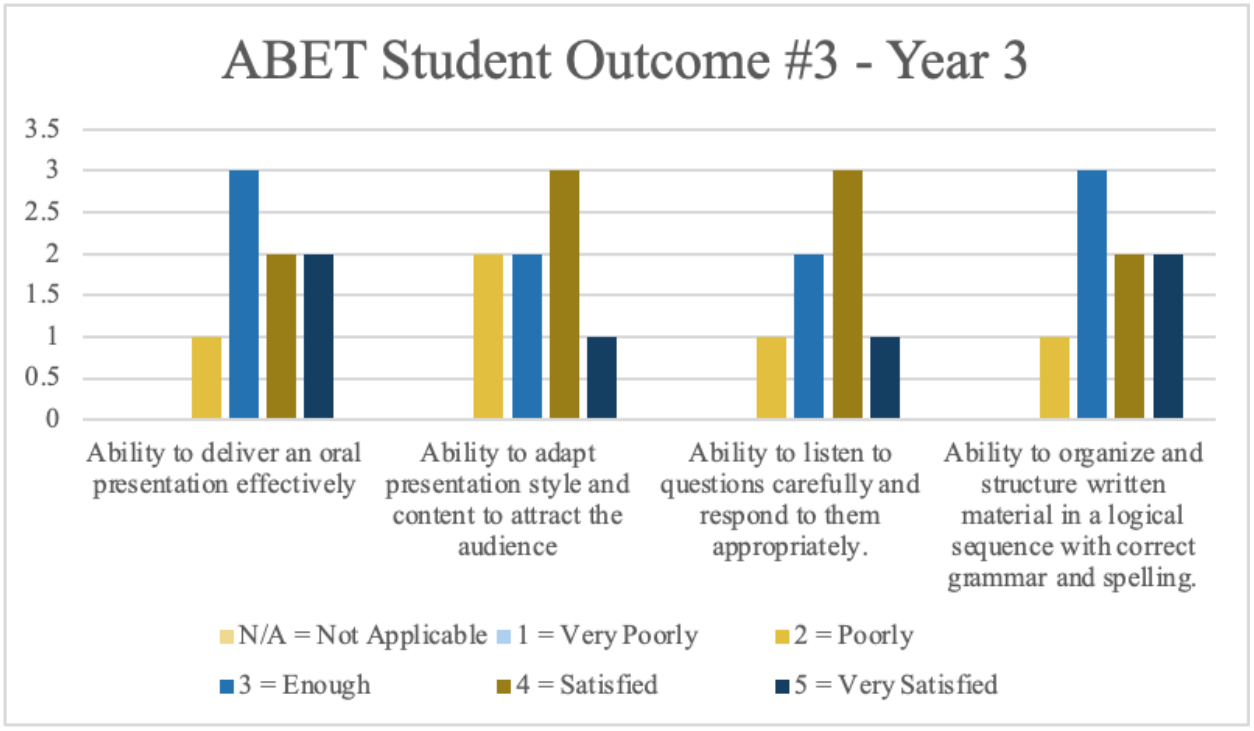

Overall Rating of Abilities in ABET Student Outcome 3



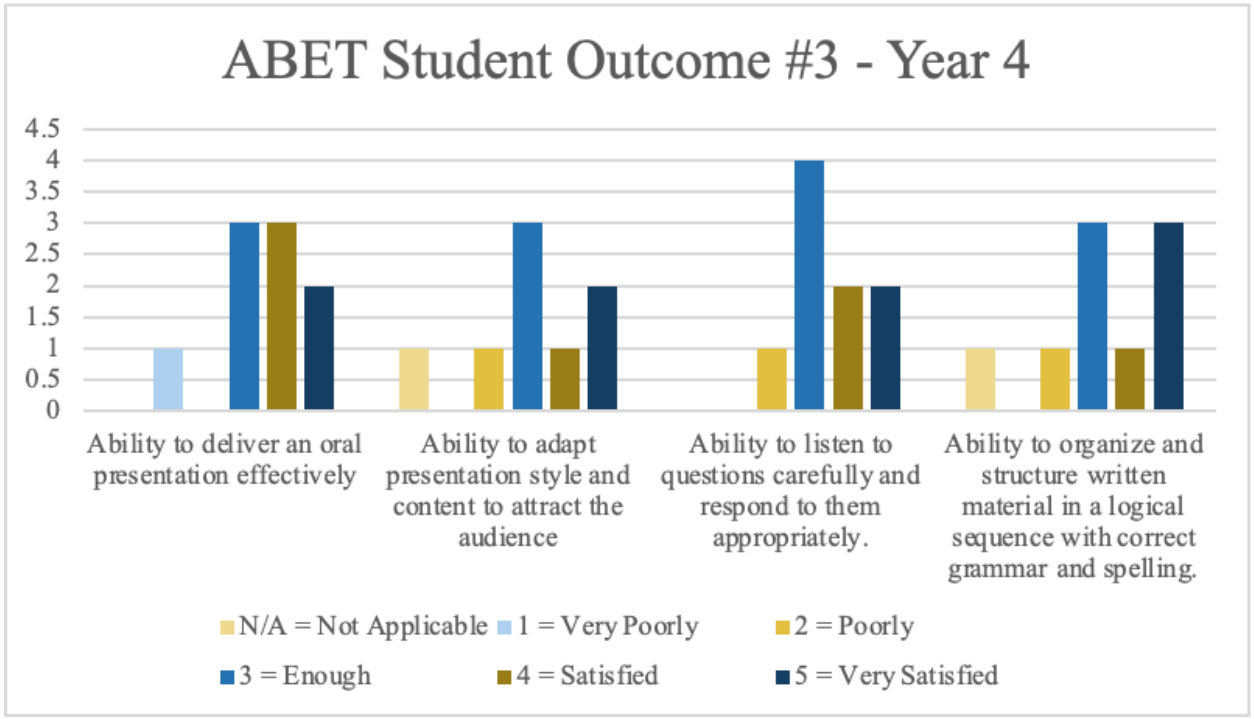

Year 1 Rating of Abilities in ABET Student Outcome 3



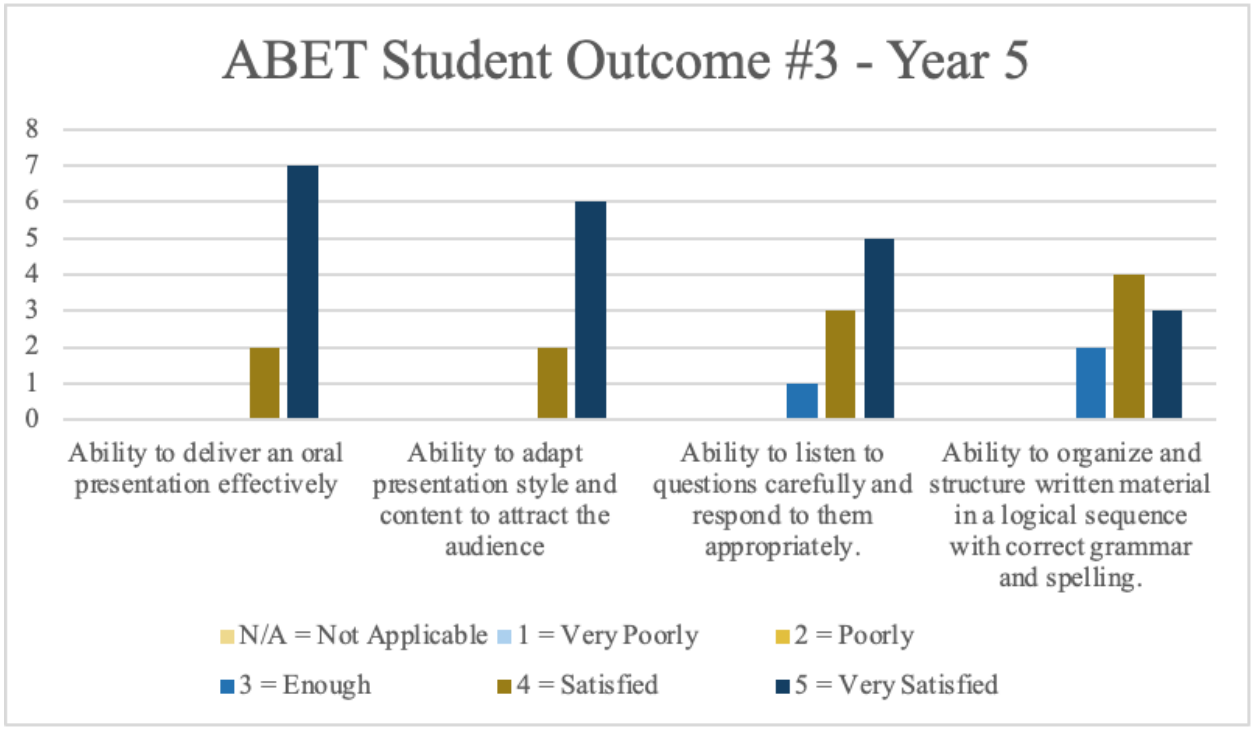

Year 2 Rating of Abilities in ABET Student Outcome 3




Year 3 Rating of Abilities in ABET Student Outcome 3

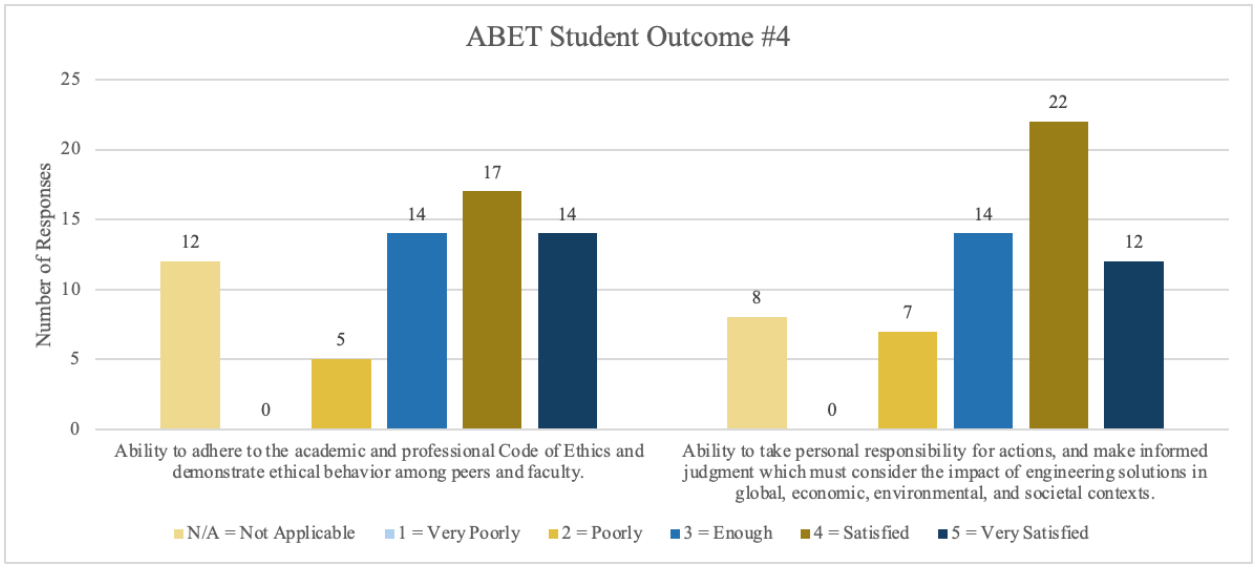



Year 4 Rating of Abilities in ABET Student Outcome 3

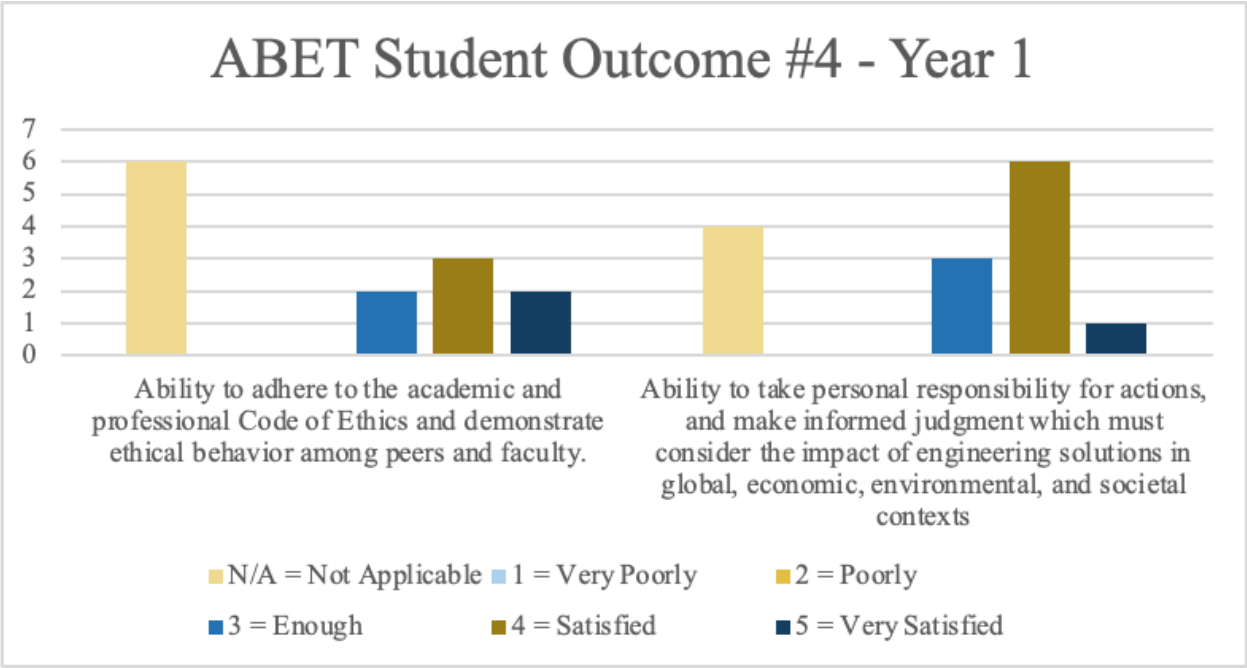



Year 5 Rating of Abilities in ABET Student Outcome 3

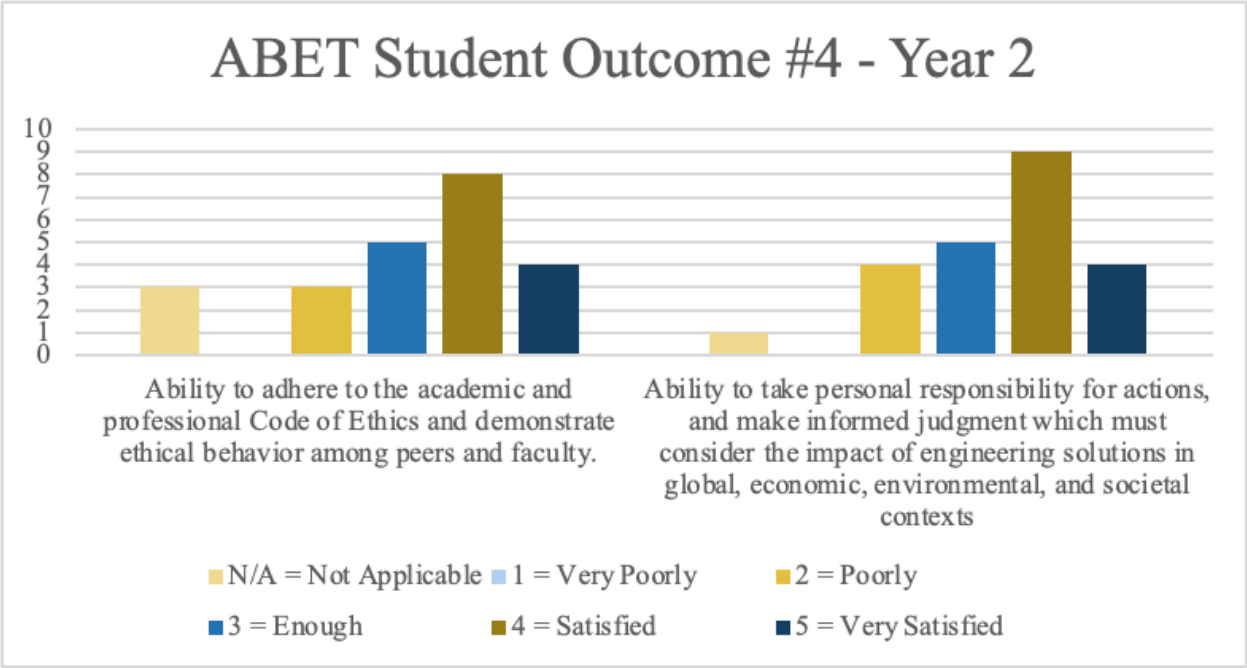
## Student Outcome #4



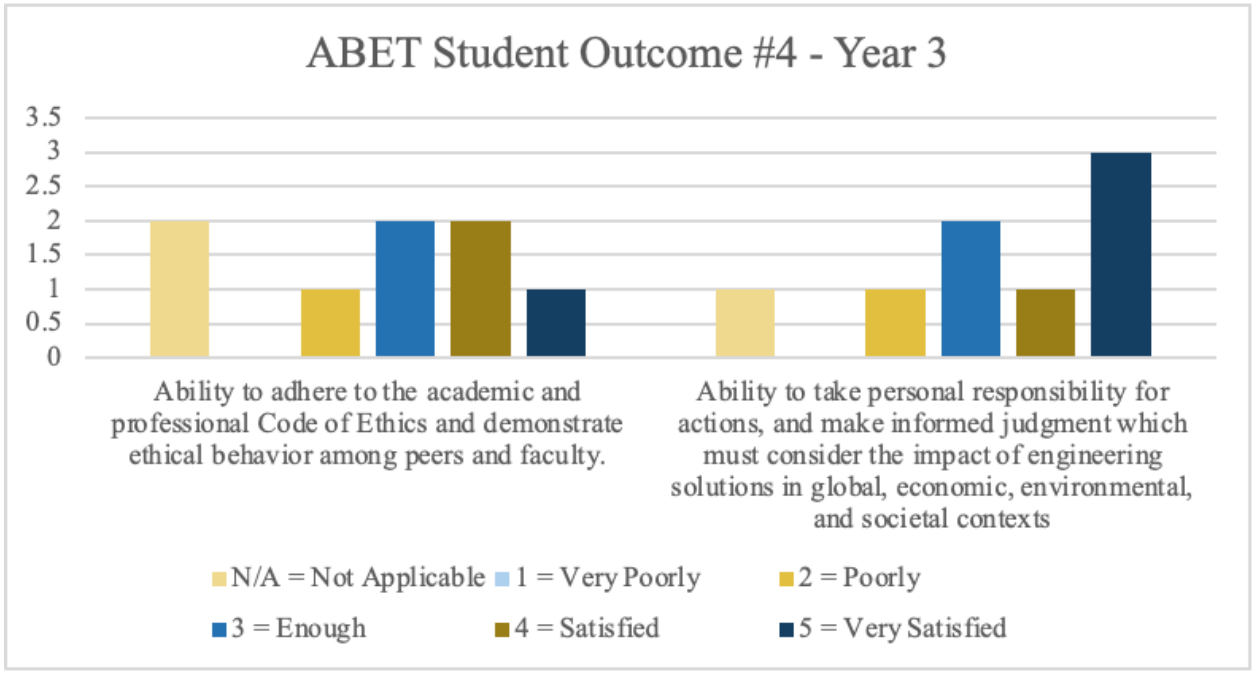

Overall Rating of Abilities in ABET Student Outcome 4



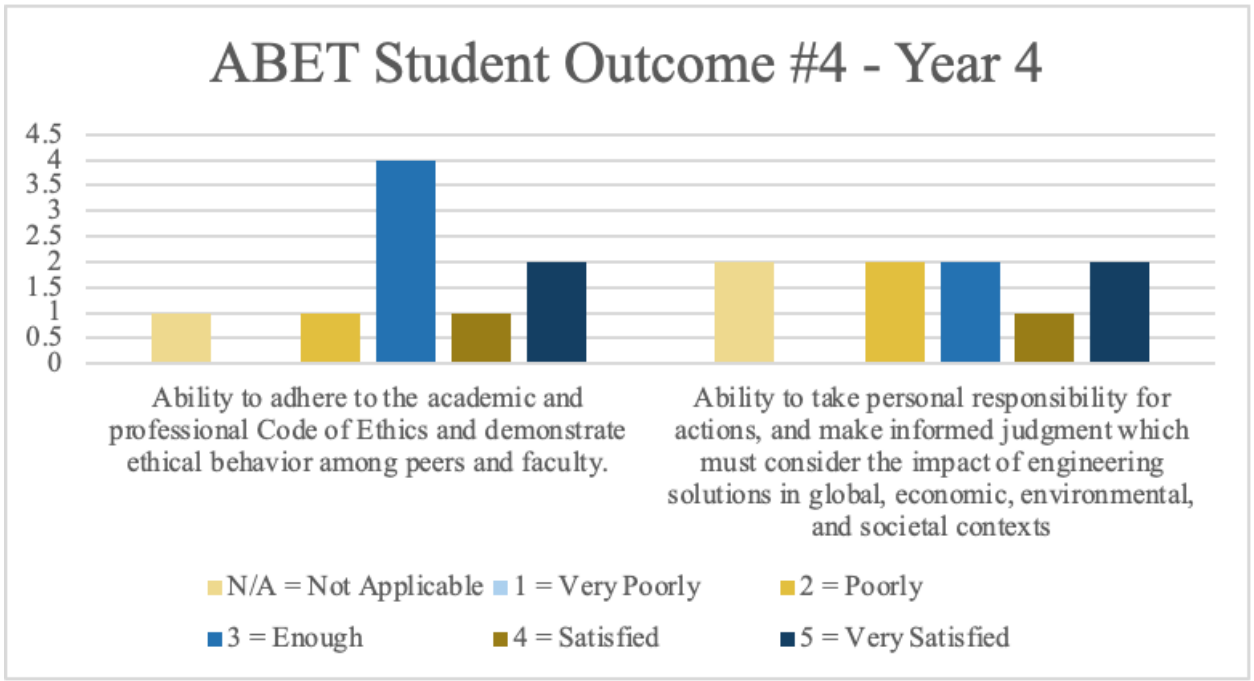

Year 1 Rating of Abilities in ABET Student Outcome 4



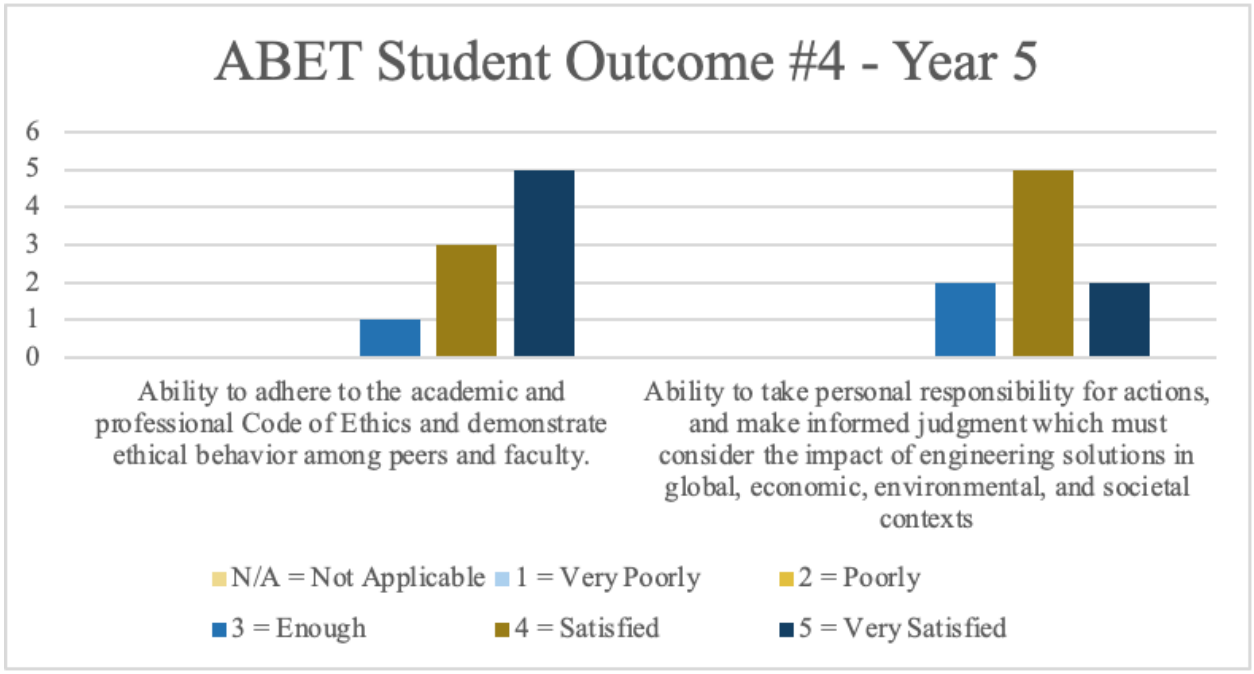

Year 2 Rating of Abilities in ABET Student Outcome 4




Year 3 Rating of Abilities in ABET Student Outcome 4

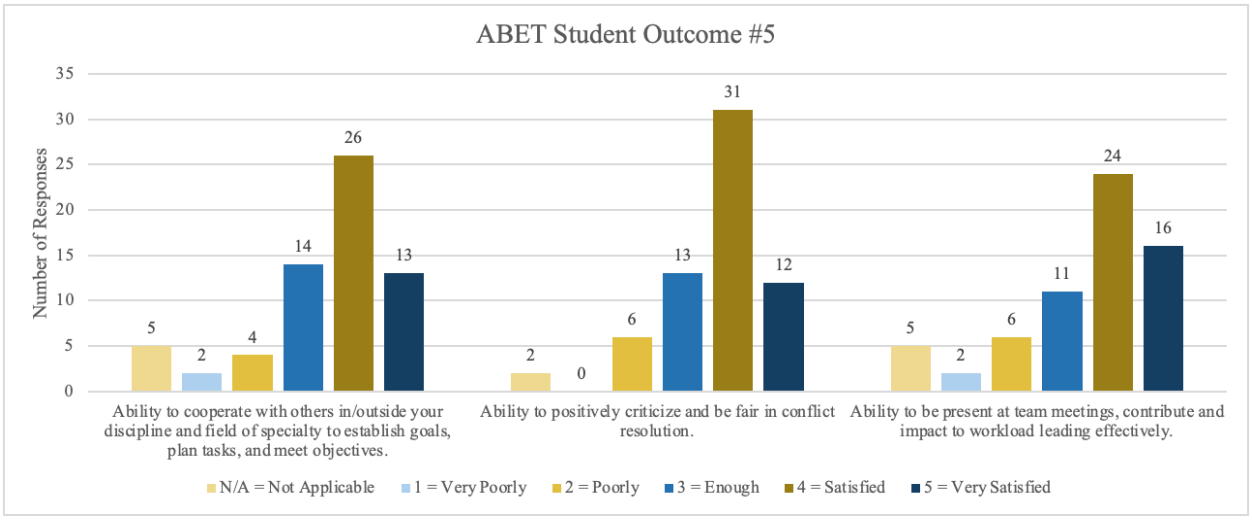



Year 4 Rating of Abilities in ABET Student Outcome 4

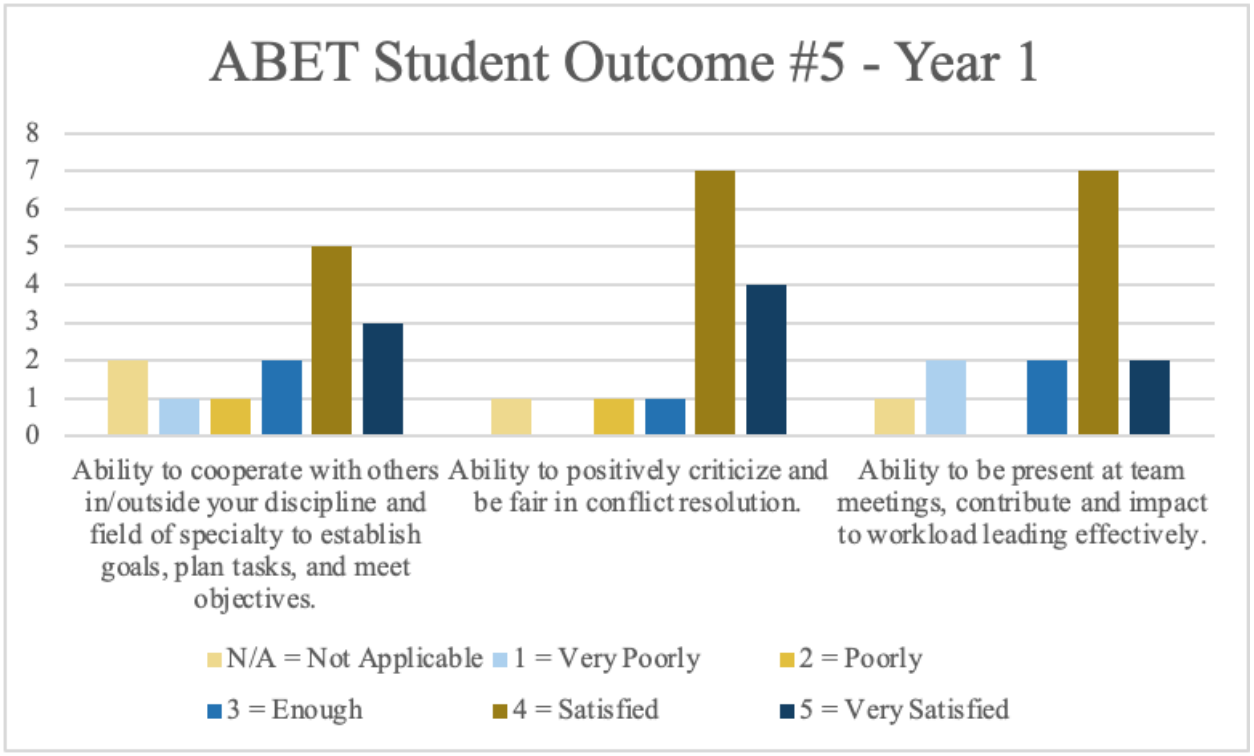



Year 5 Rating of Abilities in ABET Student Outcome 4

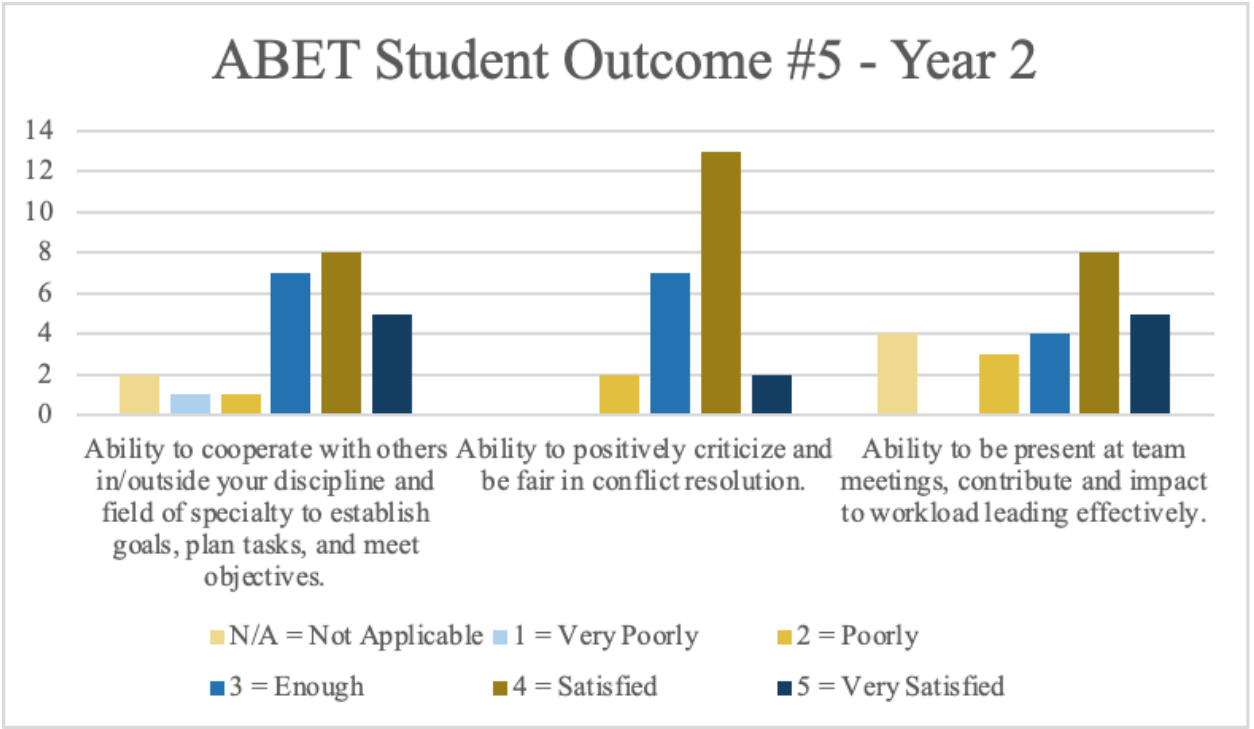
## Student Outcome #5



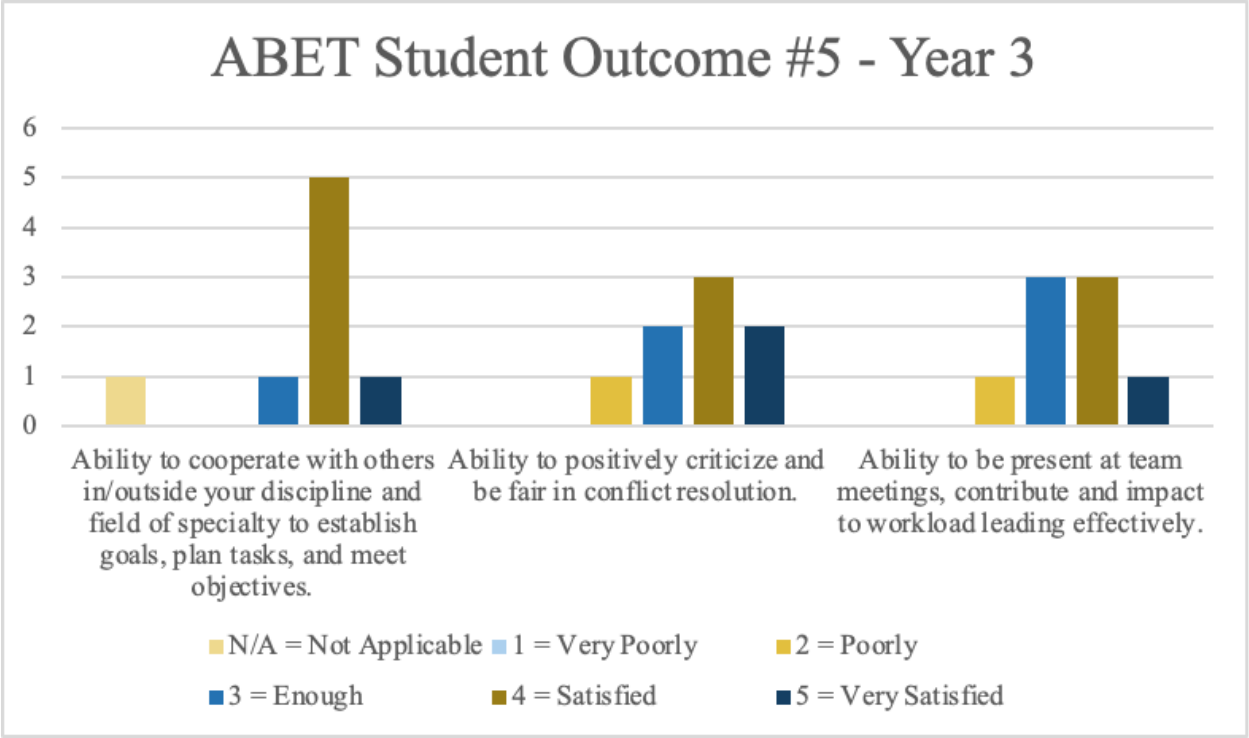

Overall Rating of Abilities in ABET Student Outcome 5



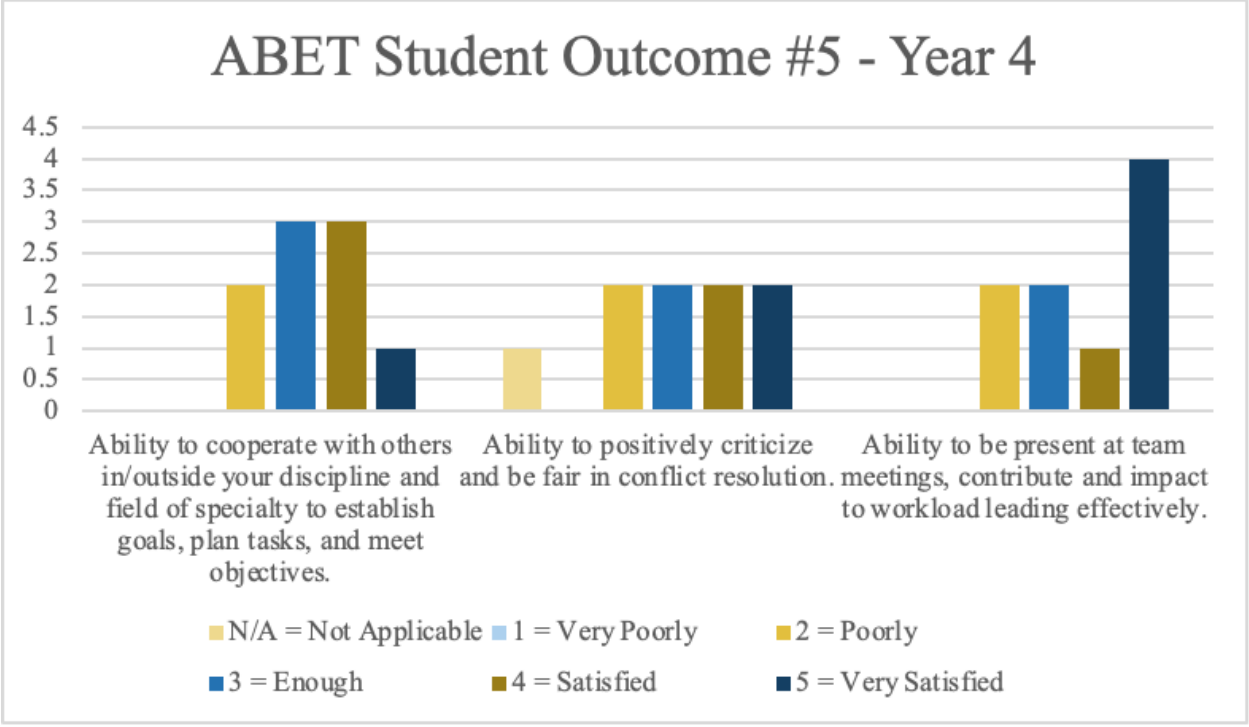

Year 1 Rating of Abilities in ABET Student Outcome 5



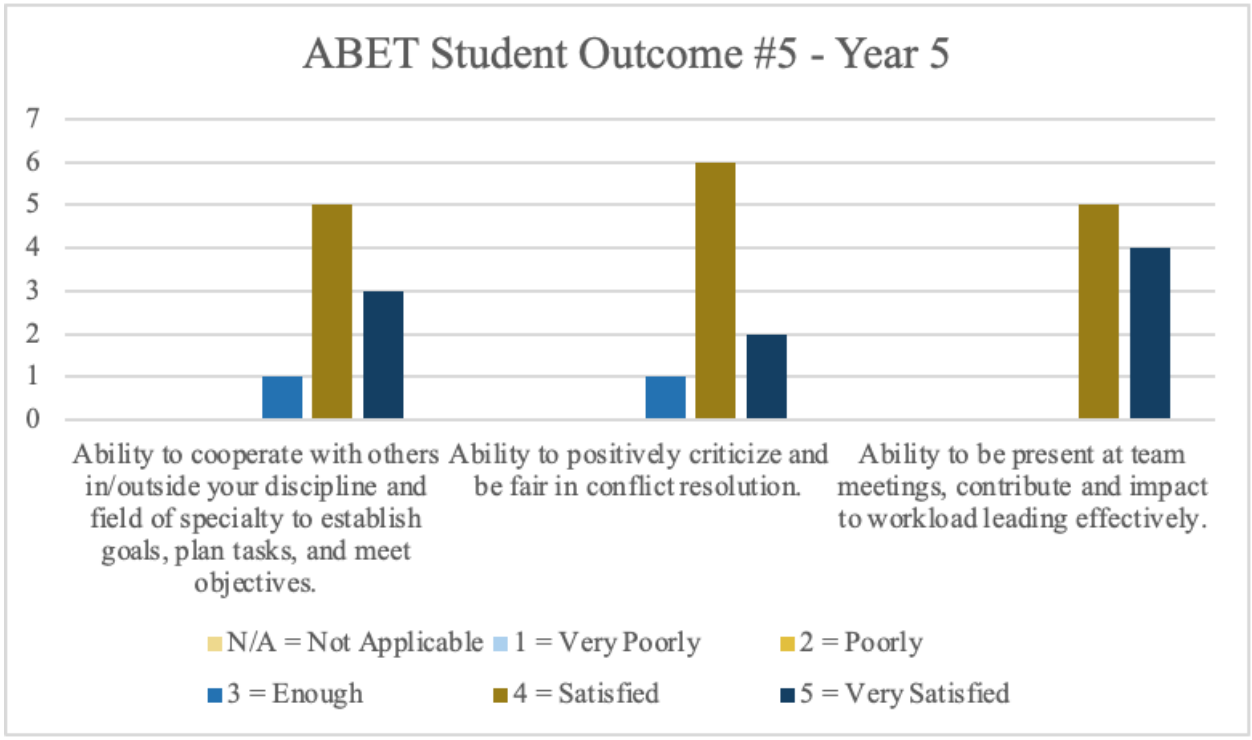

Year 2 Rating of Abilities in ABET Student Outcome 5




Year 3 Rating of Abilities in ABET Student Outcome 5

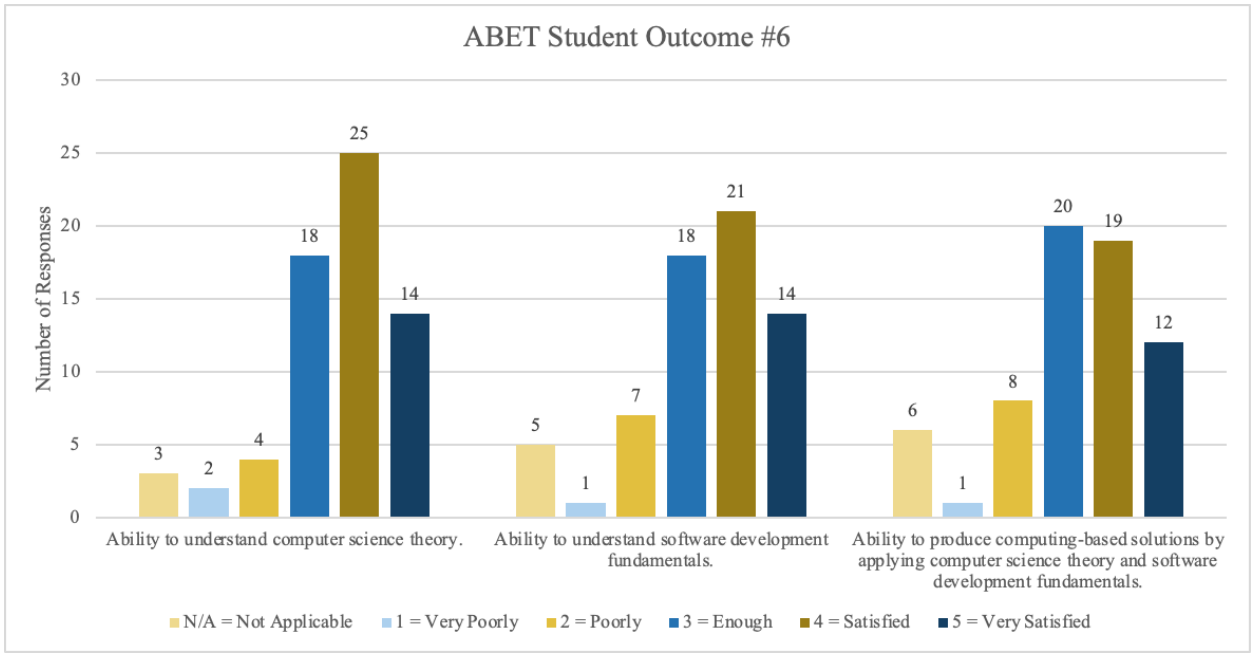



Year 4 Rating of Abilities in ABET Student Outcome 5

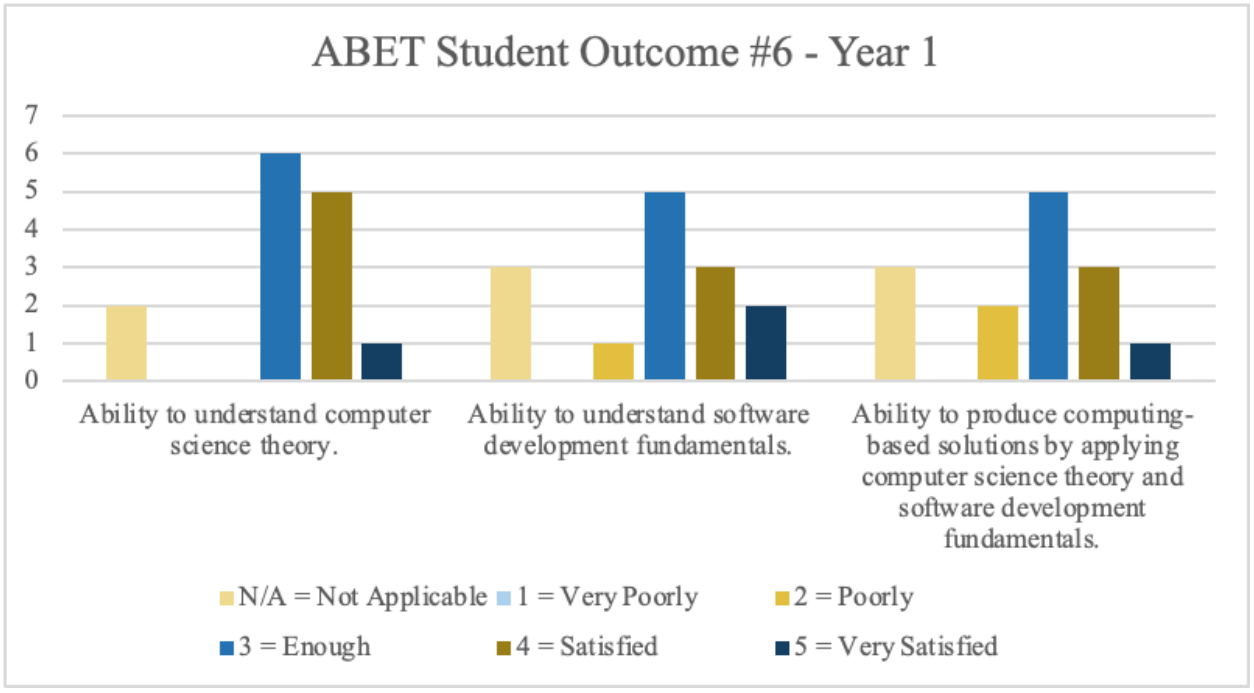



Year 5 Rating of Abilities in ABET Student Outcome 5

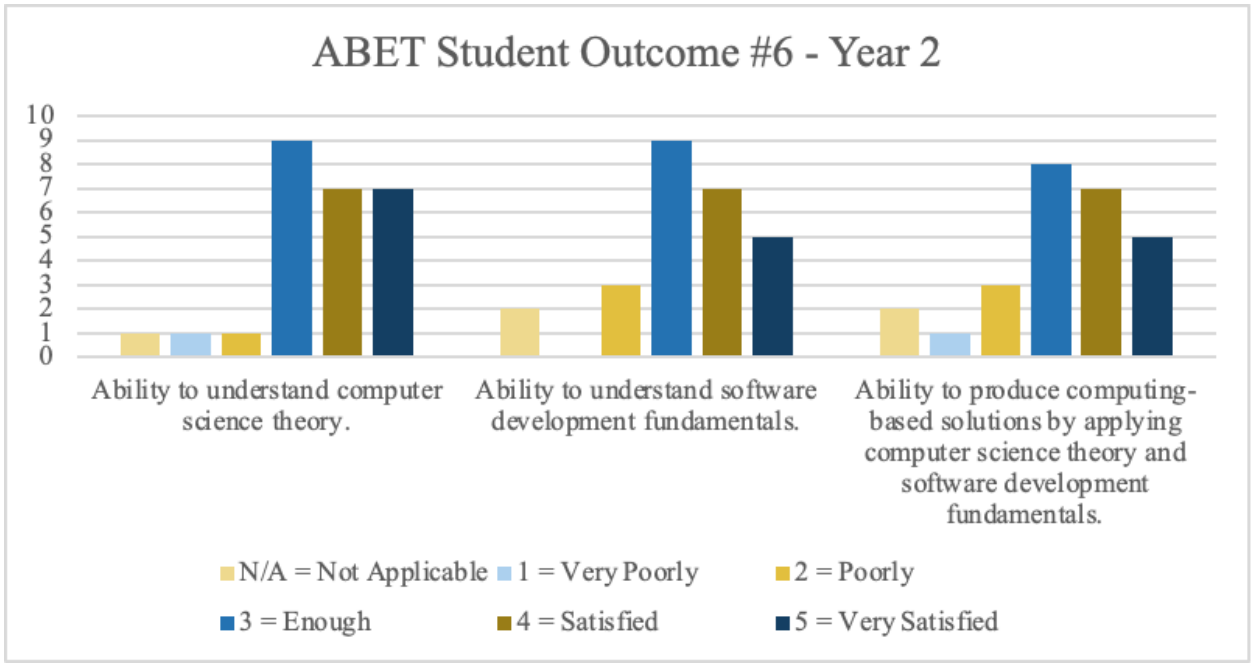
## Student Outcome #6



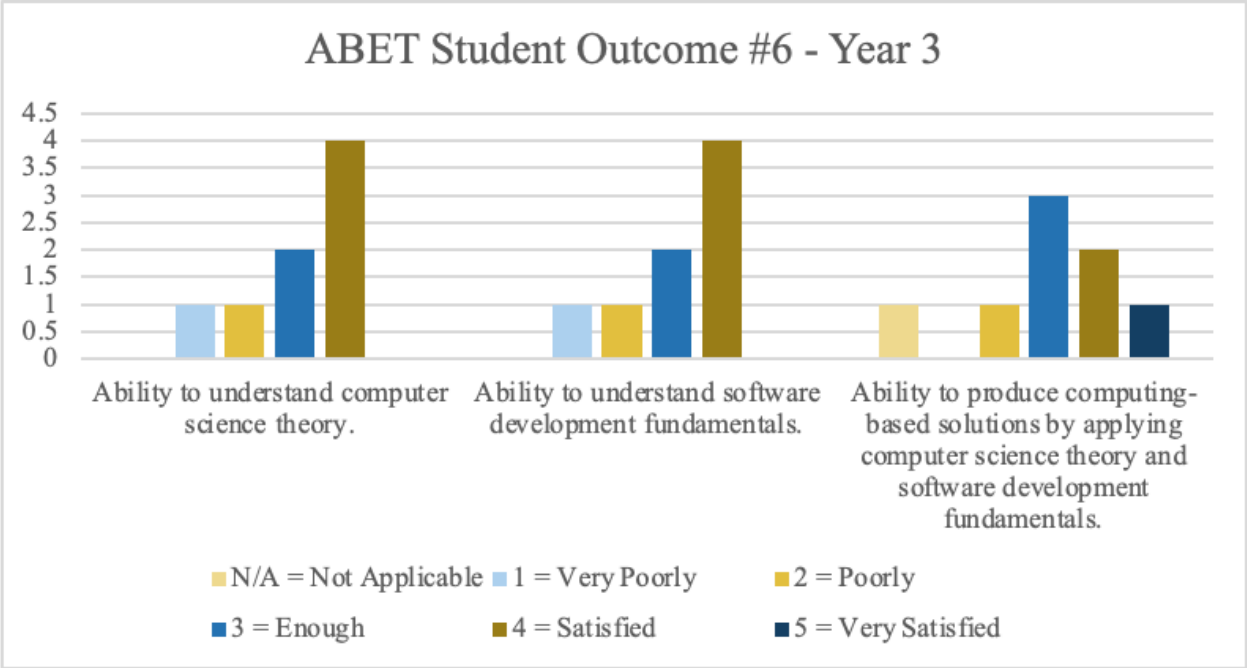

Overall Rating of Abilities in ABET Student Outcome 6



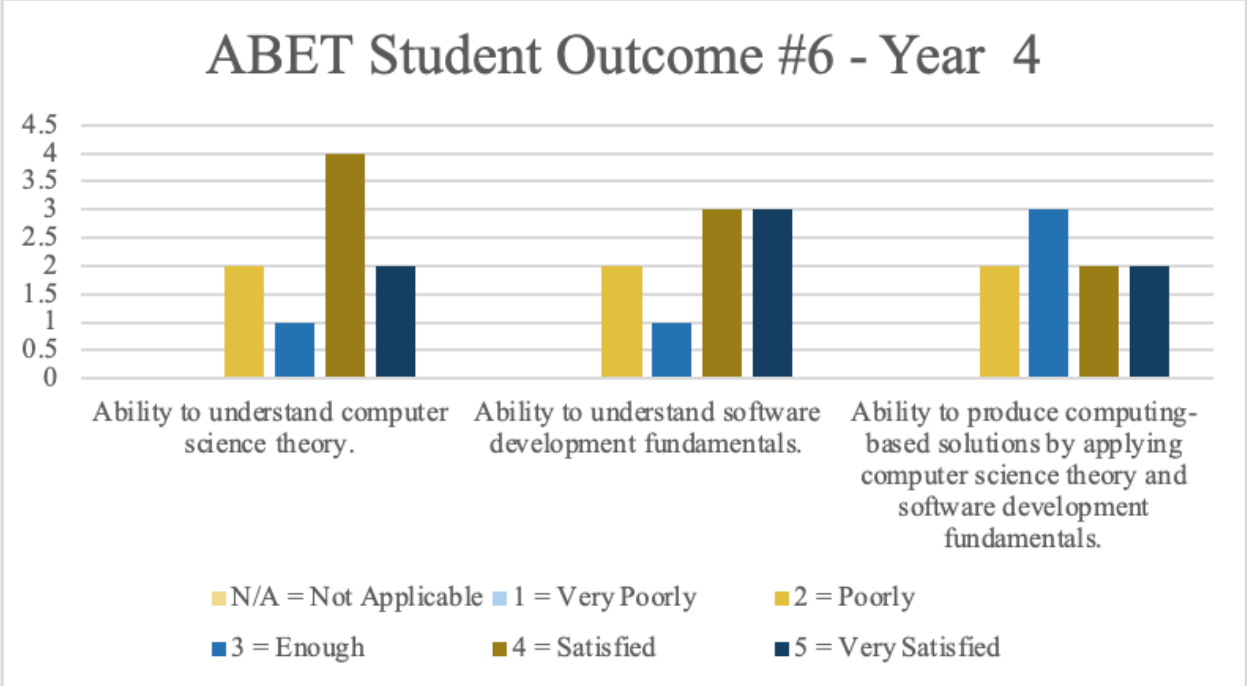

Year 1 Rating of Abilities in ABET Student Outcome 6



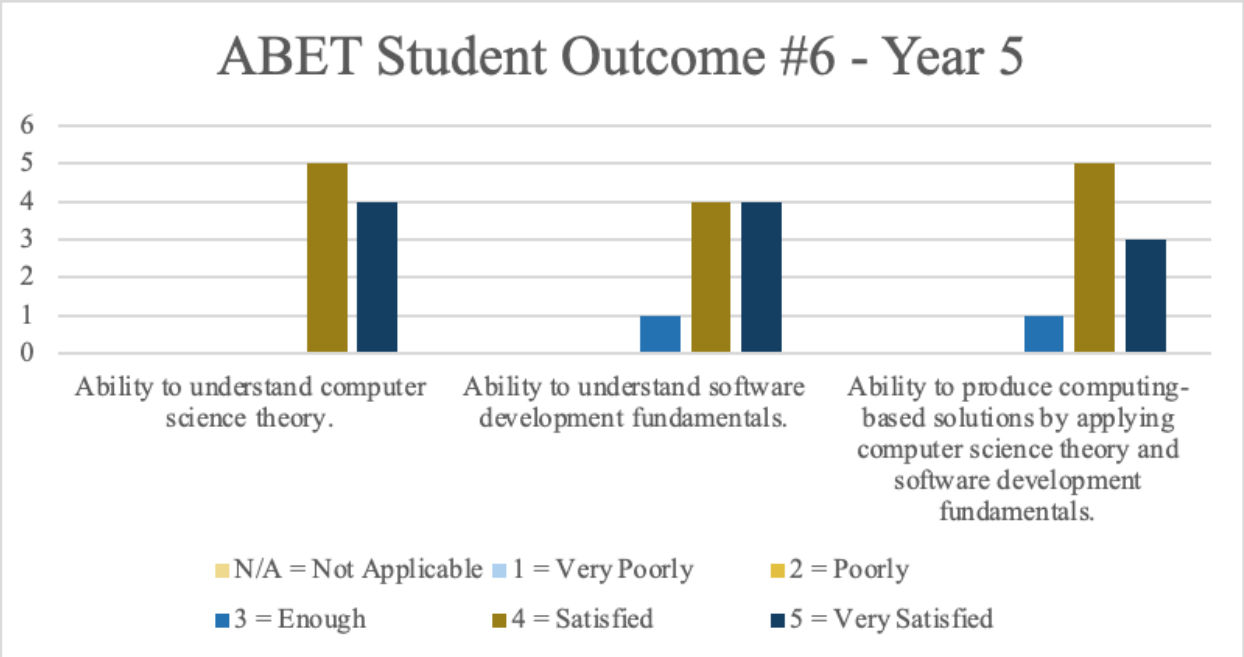

Year 2 Rating of Abilities in ABET Student Outcome 6




Year 3 Rating of Abilities in ABET Student Outcome 6




Year 4 Rating of Abilities in ABET Student Outcome 6




Year 5 Rating of Abilities in ABET Student Outcome 6