Computer Fraud and Security
ISSN (online): 1873-7056

Investigating Academic Performance Patterns among Generation Z Computer Science Students using AHP

Sami Alghamdi[1]

[1] *Computer Science Department, Al-Baha University, Al-Baha, Saudi Arabia*
samialghamdi@bu.edu.sa

Abstract

The landscape of higher education is continuously evolving, with the emergence of new generations of students bringing diverse backgrounds, expectations, and learning styles. Among these, Generation Z (Gen Z that comprising individuals born from the mid-to-late 1990s to the early 2010s and has garnered significant attention for its unique characteristics and approaches to education. As digital natives, Gen Z students are not only adept at using technology but also exhibit distinct academic preferences and challenges, particularly in fields that demand rigorous analytical and technical skills, such as computer science. This study seeks to investigate the CGPA patterns of Gen Z Computer Science students at Al-Baha University, Saudi Arabia by employing the Analytical Hierarchy Process (AHP) methodology. Overall Performance Weights for Core Course: 0.454 (Most Important), Elective Course: 0.429 (Important) and University Course: 0.142 (Least Important). The analysis reveals that while core and elective courses significantly influence CGPA, the engagement, support mechanisms, and relevance to real-world applications are critical for the academic success of Gen Z students. Educational institutions should leverage these insights by enhancing curriculum designs that cater to the needs and preferences of Generation Z, promoting practical applications, collaborative learning, and mental well-being. This holistic approach positively impacts academic performance and overall educational experience.

Keywords: Generation Z, Academic Performance, Computing Education, Analytical Hierarchy Process

1. Introduction

The world of Higher Education is continuously evolving. This new generation of students are diverse, not only in terms of background, but also in terms of expectations. They are an innovative and cutting-edge group of people, full of new ideas, and are in general very familiar with technology. Many are digital natives, and therefore have grown up with technology, in general are comfortable in using it, as well as being very familiar with a wide array of tools that are available over the internet. They have unique learning styles and ways in which they wish to be academically supported, and it is the role of the educator and institution to support them in achieving their academic potential, particularly in a discipline such as Computer Science that requires critical analytical skills as well as having high technical capability [1] – [5].

Studies on academic performance of Computer Science students in computer science field have been conducted by previous research studies [6] – [13]. The studies found that many factors affect academic performance of students studying in computer science field. These factors include previous academic performance, self-regulated learning strategies, intrinsic motivation, problem solving skills, and basic knowledge of programming. These studies will help the instructor to develop educational strategies and design learning environment and also to design the curriculum to enable students to be successful in computer science field.

The body of knowledge of studies on investigating academic performance of students studying Computer Science have been conducted by many researchers [6] – [13]. These studies explored and identified the many factors affecting academic performance of students studying Computer Science, including previous academic performance, self-regulated learning, intrinsic motivation, problem solving ability, and programming skills. The body of knowledge from these studies will be used as a basis for developing educational practices, improving curriculum, and designing strategies that could enhance the academic performance of students studying Computer Science.

Johnson and Miller [8] conducted a study which was focused on determining the intrinsic motivation of computer science students and their ability to persist in studying very difficult subjects. This study indicated that computer science students, who are studying in a computing program, are more likely to persist in studying very difficult subjects for the computer science discipline when they are intrinsically motivated to study computer science. In other words, students must be motivated by reasons that are intrinsic to study of computer science. For this to occur, students must have adequate learning experiences. Educational experiences, which students experience while they are studying computer science in a computing program, can affect the programming self-efficacy of students. As a result, the academic performance of novice programmers in a computing program is increased when the programming self-efficacy of these students is increased through appropriate learning experiences. Furthermore, a large body of research has indicated that the perceptions of learning and motivation of students are greatly affected by the degree to which students believe that all of the ideas that they present in class are heard, accepted, and taken into consideration by others in the classroom. As a result, in order to increase the motivation of computer science students in a computing program, students must feel that their ideas are taken into consideration in the learning environment of computer science.

Problem solving skills of Computer Science students studied by Brown and Black [11] in their research work is another crucial factor that can affect performance of students of different backgrounds and with different learning styles. Such skills would thus enable students of different backgrounds to perform better in different areas of study in computer science.

Other challenges include students having prior experience of programming and how this can affect their performance in their first year of university studies. The findings from a study carried out by Wilson and Taylor [12] were that students who had prior experience of programming were able to perform better than their counterparts. This would suggest that before a student enters into a university Computer Science program that they should be exposed to the basic programming concepts in a pre-requisite or a foundation course within a college or university. In addition, the factor of prior experience of programming could be taken into consideration during the university admissions process and indeed when designing the Computer Science curriculum so as to improve performance and increase chances of graduation for Computer Science students.

A host of challenges that affect learning outcomes for Generation Z students of Computer Science have recently been identified by research (Jawad & Tout, 2022). For example, rapid developments in technology or so-called 'technology-space' bring an ever-increasing array of new tools, frameworks, programming languages and associated methodologies into the 'technology-space' that cause disorientated Computer Science students a considerable number of difficulties whilst they attempt to 'keep up'. Moreover, extreme levels of anxiousness have been reported by Computer Science students who are under extreme pressure to complete their Computer Science degree successfully so that they are able to secure the best possible intern and / or the highest quality of employment following graduation. They are anxious because they are unable to earn sufficient funds from part-time employment (for example from campus restaurants, from retail outlets on and near to the campus, from odd jobs for colleagues etc.) or from student part-time employment (for example from school[s] or from colleges etc.) in order to pay off the huge student loans that they have incurred in order to study at higher education. Although huge strides have recently been made regarding growing access to technology across the world which has caused great celebration amongst stakeholders within the Computer Science community and outside of it, sadly there is also a huge 'digital gap' which exists between Great Britain and the rest of the world regarding access to the highest quality learning resources and regarding the opportunity of gaining mentors and networking opportunitys in Computer Science. The 'digital gap' affects underrepresented groups in the Computer Science community more than others. Online collaboration and communication have recently caused a large number of Computer Science students of Generation Z a great deal of difficulties. The reason for this is that a large number of Generation Z students of Computer Science have experienced difficulties in learning how to work effectively in groups and in communicating effectively with each other in both online learning environments and in hybrid learning environments. A large number of Generation Z students also experience extreme difficulties in applying theoretical concepts that were learned in CS classes to practical assignments in order to complete them successfully. In order to deal with the challenges that affect Generation Z students of Computer Science, it is necessary for Computer Science educators and for institutions of higher education and also for industry to implement a number of solutions to a number of problems. For example, a review of current gamification techniques with university–level case studies of CS students, a long-running study into a number of reforms to pedagogy which are particularly suited to 'digital native' Computer Science students of Generation Z, current and future uses of ChatGPT by students of Gen Z from around the world, a pilot study into the use and the 'limits' of voice-automated systems, or voice assistants, for example with examples from CS students of Generation Z who studied Computer Science at university.

A lot of efforts from educators, educational institutions and industries are needed to support Gen Z students in Computer Science. A lot of studies have been conducted to understand academic performance of Gen Z students in Computer Science program in order to improve their learning experience and learning environment and to design better educational strategies and learning support for them. In order to measure academic performance of students in university, the most primary and sufficient metric to use is the average grade of students in all of the courses they have taken during their time in university. This average grade is expressed in the form of students' cumulative grade point average (CGPA). Although using the average CGPA of students in university provides sufficient information on academic performance of students in university, it does not provide information on different types of courses that students took in university such as university-wide courses, college-wide courses and department-wide courses. Thus, this study aims to identify and analyze performance of boys in undergraduate program of Computer Science at college of Computing and Information, Al-Baha University of intakes 2018, 2019, 2021 and 2022. The study will rely on all available data for the batches of interest.

This study is an attempt to investigate the CGPA patterns of the Gen Z students studying Computer Science at the Al-Baha University, Saudi Arabia. To achieve the objective of the study, AHP methodology will be used to evaluate the relative importance of different course types offered to the students studying Computer Science. AHP provides a systematic approach to establish relative importance of different criteria, in this case different course types, and their impact on CGPA of the students. The methodology enables the use of data- comparison to develop prioritized strategies for improving effectiveness of the different course types. The integration of AHP-TOPSIS with Long Short-Term Memory (LSTM) machine learning technique has been used for the evaluation and prediction of performance of higher education institutions. This study will thus help in the management and development of higher education in Saudi Arabia. The proposed approach enables a comprehensive analysis [22] of different aspects of the performance of students and thus will enable the instructor to develop appropriate strategies for instruction and student services. The objectives of this research study are to:

1. Analyze the academic performance distribution of Gen Z Computer Science students across different course types.
2. Assess and rank the impact of various course types on CGPA of different levels of students through the AHP methodology.

This study attempts to answer the above-mentioned questions. It attempts to build on the body of research that has already been carried out in order to understand the learning and university experiences of the Gen Z university students. This study, in particular, focuses on the Computer Science students at the college of Computing and Information at Al-Baha University. The rest of this paper is organized into four main sections. In Section 2, a detailed description of the proposed method that is to be used for the analysis of the performance of the Gen Z CS students as well as for the evaluation of the various educational strategies that are currently being used at the college of Computing and Information at Al-Baha University is presented. The results of the analysis of the performance of the Gen Z CS students at the college of Computing and Information at Al-Baha University are presented in Section 3. The results include the performance distribution of the Gen Z CS students as well as the ranking of the impact of the different course types on the average CGPA of the Gen Z CS students as well as the assessment of the impact of the different levels of students on their average CGPA. Finally, the conclusions of this study as well as the possible future work are presented in Section 4.

2. Material and Method

The Analytical Hierarchy Process (AHP), which was introduced by Saaty [19], is a method of multi-criteria decision making that uses weights which are determined by a set of pairwise comparisons between elements of a hierarchy that consists of a goal, criteria, subcriteria and alternatives. The AHP hierarchies that are used in the presented research are shown in Figure 1. The highest level of AHP hierarchy (Level 3) contains the alternatives which represent the student's cohorts. Three categories for the average performance of the courses by the student's cohorts are considered: weak, mediocre and excellent. The criteria that are considered in the research are: average performance of University courses, Core courses and of the Elective courses, as well as the prior GPA of the student. The hierarchy consists of three levels. The highest level (Level 1) contains the most influenced factor for the academic performance of a student within a cohort of university students. The second level (Level 2) contains the criteria that are considered in the research. The third level (Level 3) contains the alternatives that represent the student's cohorts in the university. Four different alternative cohorts of students are considered: Freshman, Somophore, Junior and Senior. The Core courses are compared with the Elective courses on the subcriteria level for each of the three criteria.

- Top (goal): Explain/weight factors that correlate with student achievement across cohorts.
- Criteria: University course, Core course, Elective course, prior GPA.
- Subcriteria: core vs elective.
- Alternatives: the cohorts or the student groups

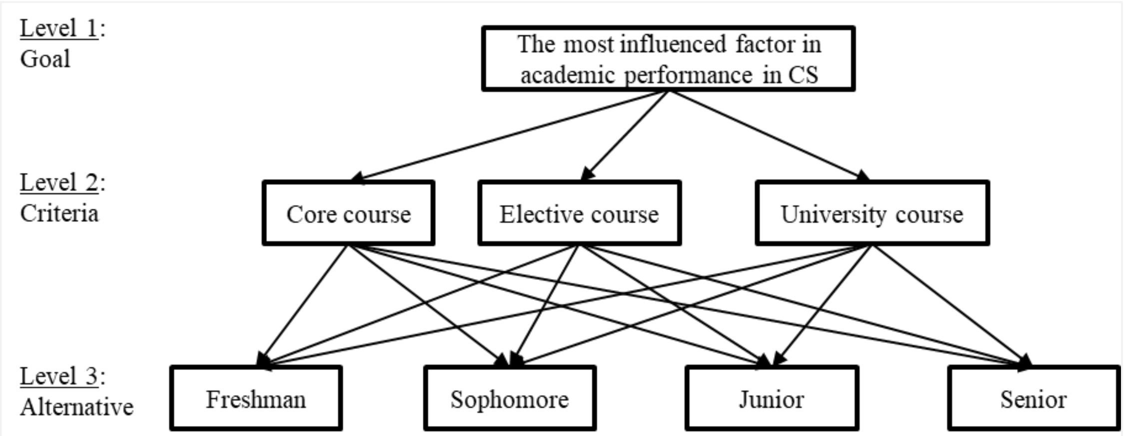

Figure 1. AHP Hierarchy for the proposed method

2.1 Data Preprocessing

Average grade points of academic performance are calculated for all different course types (Core, Elective, University) for all different cohorts of students (Freshman, Sophomore, Junior, Senior). The average performance of a student with all his/her different course types is sorted in an increasing order and then receives the label of being Weak, Mediocre or Excellent by his/her average grade score. The grade score average is categorized according to the rules below: Weak – Grade score average ≤ 70; Mediocre – 70 ≤ Grade score average < 85; and Excellent – Grade score average ≥ 85.

2.2. AHP Algorithm

The AHP method has been applied in this research to assess the impact of different course types on the academic performance of Generation Z students who are studying Bachelor of Computer Science program at university. AHP is a method for decision making. In this method, factors that are important for a decision are rated, and a hierarchy is constructed to represent these factors. Also, priority weights are determined for the factors in the hierarchy. In this research, the hierarchy consists of the influence of Core, University compulsory support, and Elective courses on the academic performance of students measured by their CGPA as described in Algorithm AHP.

AHP Algorithm Steps

- Input and Output
  - Goal: Evaluate the impact of course types on CGPA.
  - Criteria: University Course (U), Elective Course (E), Core Course (C)
  - Output: Pairwise Comparison Matrices, Priority Vectors and Consistency Ratios

- Data Preprocessing: Determine
  - Grade score average for each student at each level.
  - Students' category.
  - Correlation coefficient (ρ) among criteria variables using (1), *n* is the amount of data.

$$\rho_{x,y} = \frac{\sum xy - (\sum x \sum y)}{\sqrt{(n\sum x^2 - (\sum x)^2)(n\sum y^2 - (\sum y)^2)}} \quad (1)$$

- Pairwise Comparison Matrices

To make sense of the correlation differences, a mapping function called PC(R) was used. This function helped assign comparison values based on how strong the correlations were, as shown in equation (2) [15]. Here, ρc refers to the correlation for a Core Course, ρe is for an Elective Course, and ρu represents the correlation for a University Course. These correlations are important to understand the relationships between different courses.

$$\begin{aligned} PC(R) &= 1 \; if \; \left|\rho_i - \rho_j\right| \leq 0.1 \text{ (Equal importance)} \\ &= 3 \; if \; \left|\rho_i - \rho_j\right| \leq 0.3 \text{ (Moderate importance)} \\ &= 5 \; if \; \left|\rho_i - \rho_j\right| \leq 0.5 \text{ (Strong importance)} \\ &= 7 \; if \; \left|\rho_i - \rho_j\right| \leq 0.7 \text{ (Very strong importance)} \\ &= 9 \; if \; \left|\rho_i - \rho_j\right| \leq 0.9 \text{ (Extreme importance)} \end{aligned} \quad (2)$$

- Weights Calculation
  1. Normalize the entry in the pairwise comparison matrix using (3).

$$N_{ij} = \frac{a_{ij}}{\sum_1^n a_{ij}} \quad (3)$$

  2. Calculate the Priority Vector w using (4).

$$w_i = \frac{1}{n}\sum_1^n N_{ij} \quad (4)$$

where *n* is the number of criteria being compared.

- Calculate Consistency Index (*CI*) and Consistency Ratio (*CR*) using (5) and (6)

$$CI = \frac{\lambda_{max} - n}{n-1} \quad (5)$$

$$CR = \frac{CI}{RI} \quad (6)$$

where λ_{max} is the maximum eigenvalue of the matrix and *RI* is the Random Index based on the number of criteria.

3. Results and Discussion

In this section, the results of research are explained along with the comprehensive discussion. The results of data preprocessing are presented in Section 3.1, followed by the AHP results in Section 3.2. Section 3.3 provides a comprehensive discussion. The proposed AHP structure is implemented using Python programming language on a high performance computer with ntel Core i9-14900K CPU, NVIDIA RTX 5080 GPU, and 64GB of DDR5 RAM specification.

3.1 Data Preprocessing Results

Academic performance data are taken from learning management system of Al-Baha University, considering cohort from 4 batches: 1438H, 1439H, 1441H, 1442H (The university's academic year is based on Hijri lunar calendar). The dataset includes historical academic records for 107 students spanning 17 semesters, combined with details from their study plans. It comprises 5,101 course-grade entries from 51 distinct courses distribute into 4-year program as depicted in Figure 2. Table 1 shows the credit hours (CHs) of the study plan of the program. Figure 3 shows a snippet of the dataset contents to illustrate the structure of the record data. As we can see, the University Courses are offered during the first and second year, however, many students deffer taking the courses towards their final year of ther study. The defferment somehow affecting the academic performance for some students. The Elective courses are offered during year 3 and year 4, however, there are transfered students from other colleges/ universities and usually they are allowed to take the Elective course earlier. The Core courses are offered all over the year.

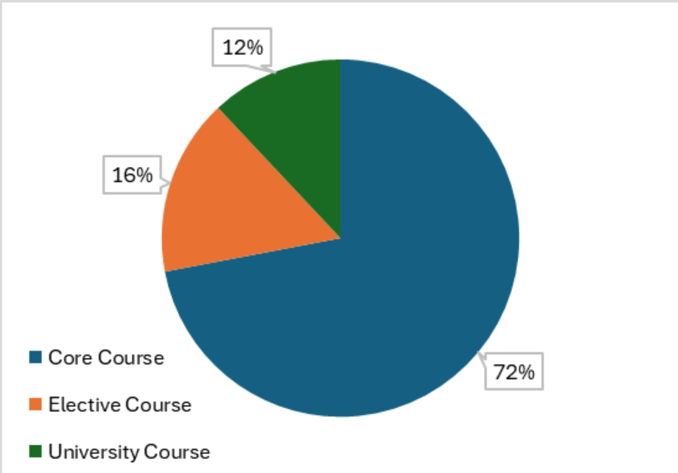

Figure 2. Distribution of the number of courses in the program

	SEMESTER_ ID	STUDENT_ ID	COURSE_ CODE	COURSE_TITLE	GRADE_ LETTE	GRADE_ SCOR
2	201801	[redacted]225	CS10301	Advanced Programming	B	82
3	202001	[redacted]225	CS10004	Advanced Web Applications	B	81
4	201701	[redacted]225	MATH1000	Calculus 1	A+	95
• • •						
5100	202201	[redacted]682	CS10503	Software Engineering	D	63
5101	202001	[redacted]682	CS10303	Statistics and Probability	D	60
5102	202101	[redacted]682	CS10401	Web Page Development	D+	66

Figure 3. Snippet of the dataset

Crse_Type	Year offered			
	Year 1	**Year 2**	**Year 3**	**Year 4**
Core	20 CHs	33 CHs	33 CHs	27 CHs
Elective	-	-	6 CHs	18 CHs
University	12 CHs	4 CHs	-	-

Table 1. Credit hours distribution of the program study plan

		Freshman				Sophomore				Junior				Senior				
No.	**S_ID**	**C-Score**	**E-Score**	**U-Score**	**CGPA**	**C-Score**	**E-Score**	**U-Score**	**CGPA**	**C-Score**	**E-Score**	**U-Score**	**CGPA**	**C-Score**	**E-Score**	**U-Score**	**CGPA**	**C-Score**
1	**519	86.3	-	82.4	3.3	85.5	-	85	3.5	92.3	91.5	90	3.8	95.1	95.5	-	4	92.3
2	**225	84.5	-	73.2	3.0	79.5	-	85	3.3	83.2	86.5	86	3.6	85.7	86	-	3.5	83.2
3	**302	59.5	-	86	2.2	77.6	-	83	2.9	83	85	86	3.5	82.9	84.3	-	3.4	83
4	**932	84.8	-	93.7	3.6	91.1	-	93	3.8	90.5	99		3.9	88.9	95	-	3.7	90.5
5	**471	76.3	-	87.6	2.9	70	-	77	2.4	76	83	85	3.0	75	72.5	-	2.5	76
…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
107	**682	-	-	-	-	74.8	80	70	2.7	71	-	-	2.0	66.8	73.3	-	2.0	71

Table 2. Examples of Grade score average calculation results

	Year 1	Year 2	Year 3	Year 4	Overall
$\rho_{c,u}$	0.55	0.34	0	0.16	0.85
$\rho_{c,e}$	0	0	0.81	0.95	0.6
$\rho_{e,u}$	0	0	0	0.4	0.35

Table 3. Correlation coefficient of the three criteria

Table 2 shows the results of average grade scores calculaton for each student as Freshman (year 1), Sophomore (year 2), Junior (year 3), and Senior (year 4). The scores are grouped into 3 different types of courses: University (U-Score), Core (C-Score), and Elective courses (E-Score). The table also shows the category of students according to the grade score achievement. The scores are fed into AHP algorithm. Table 3 shows the correlatation coefficients between each criterion against the CGPA. The correlation coefficient for each criterion was calculated as follows: ρ_c = 0.93, ρ_e = 0.88, and ρ_u = 0.4.

3.2. AHP results

After carrying out the calculations using the AHP algorithm, we summarize our findings. Table 4 shows the Pairwise comparison matrices calculation, Table 5 shows the weights calculation results.

The maximum Eigenvalue (λ_{max}) is calculated based on the matrix, i.e.: λ_{max} = 3.05 and the number of criteria, n=3. The CI is calculated using (5) as follows.

$$CI = \frac{\lambda_{max} - n}{(n-1)} = \frac{3.05 - 3}{3 - 1} = 0.025$$

The CR is calculated using (6), where RI is the random index, which is based on the number of criteria. For (n = 3), the RI = 0.58, thus,

$$CR = \frac{CI}{CR} = \frac{0.025}{0.58} = 0.0431$$

The calculated CR of approximately (0.0431) is below the generally accepted threshold (typically 0.10) for acceptable consistency in pairwise comparisons. This result indicates that the judgments made in the pairwise comparisons are consistent.

Criteria	Year 1			Year 2			Year 3			Year 4			Overall		
	C	E	U	C	E	U	C	E	U	C	E	U	C	E	U
C	1	1	5	1	1	3	1	7	1	1	9	1	1	3	3
E	1	1	1	1	1	1	1/7	1	1	1/9	1	3	1/3	1	5
U	5	1	1	3	1	1	1	1	1	1	1/3	1	1/3	1/5	1

Table 4. Pairwise comparison matrices result

Criteria	Weight				
	Year 1	Year 2	Year 3	Year 4	Overall
C	0.526	0.682	0.775	0.552	0.454
E	0	0.017	0.225	0.402	0.425
U	0.484	0.301	0	0.086	0.121

Table 5. Weights (Priority factors) calculation result

3.3. AHP result analysis

Based on the AHP analysis of the impact of course types (University Course, Core Course, and Elective Course) on CGPA, the findings provide valuable insights into the academic performance of Computer Science students across different study years (Year 1 to Year 4). As shown in Table 5, overall Performance Weights for Core Course (C): 0.454 (Most Important), Elective Course (E): 0.429 (Important) and University Course (U): 0.142 (Least Important). Additional insights on cohort analysis are described as follows.

- Year 1: Students generally are transitioning into higher education, often forming their academic identities. The importance of core courses, which build fundamental programming and analytical skills, is reflected in their significant impact on CGPA. Gen Z students, known for their digital literacy and multitasking capabilities, may excel in engaging course material but could benefit from additional support in adapting to college-level rigor.
- Year 2: As students become more acclimated, performance in university courses becomes critical. The close values of core courses and university courses indicate that foundational knowledge combined with practical application leads to improved CGPA. Gen Z's preference for collaborative and interactive learning might enhance their understanding of university courses, fostering performance.
- Year 3: Elective courses have a lower weight, suggesting they contribute less to overall performance compared to core and university courses. However, this cohort may also start to specialize in areas of interest, aligning with Gen Z's desire for personalized learning experiences. The focus on core and university courses is essential as students start seeing the relevance of their studies for their future careers, which is paramount for Gen Z's career-driven mindset.
- Year 4: By the final year, students are likely integrating their learning experiences from all course types. The lower weight attributed to elective courses indicates they may not have as significant an impact on CGPA, as students focus on completing core requirements and university projects. Gen Z often seeks real-world applications for their skills, and integrating internships or industry projects could enhance their academic performance significantly.

In addition, generational insights on Gen Z Computer Science are elaborated as follows.

- Digital Natives: As digital natives, Gen Z students look for interactive and technology-enhanced learning experiences. The structured nature of core courses often provides them with the foundational knowledge needed to thrive in their digital engagements.
- Preference for Practical Application: Gen Z values hands-on learning and practical applications. The importance of university courses mirrors their desire for learning that is relevant to their future careers, leading to sustained motivation and higher performance.
- Collaboration and Community: Reflecting a strong preference for collaboration, Gen Z students typically perform better in academic settings that foster teamwork. This insight suggests the need for educational environments that promote peer interaction, especially in core and university courses, to support academic performance effectively.
- Focus on Well-being: Mental health and well-being are paramount for Generation Z. The increased pressure to excel academically can lead to stress, particularly in core courses. Support systems and resources to aid in managing academic stress will be crucial for enhancing student performance across cohorts.

Thus, based on these insights, university as a responsible institution should focus on creating a holistic educational framework that addresses the needs of Generation Z students. By enhancing curriculum, providing robust support systems, tapping into technology, and fostering collaborative environments, educational institutions can significantly improve academic performance and prepare students for successful careers in a rapidly evolving workforce. This initiative will not only empower students but also contribute to the broader educational mission of developing skilled and adaptable professionals.

3.4 Student distribution based on cohort

Observing the student distribution with regards to cohort, the trend is positive as shown in Table 6. In Freshman level, students need to adapt to new study environment, however, the number of students in Weak category decrease from year to year. Few students delay the University courses until their final year, and with this strategy, they improve the performance.

Level	Student Distribution		
	Weak	Mediocre	Excellent
Freshman	48	37	22
Sophomore	42	32	33
Junior	36	35	36
Senior	20	42	45

Table 6. Student distribution based on cohort

4. Conclusion

The results of the assessment into the different types of programs offered at Helix Institute and how they affect the CGPA of Generation Z students who are studying Computer Science revealed core and elective programs both have significant effects on the CGPA of Generation Z students; more importantly the aspect of engagement, support and relevance in academic programs are key determinants to the success of Generation Z students in academic programs and with the implementation of a few key steps Generation Z students can be provided with effective learning experiences through quality academic programs.

Based on the findings for the Computer Science students, future work will likely be focused on the following topics:

1. Enhanced Practical Application through Core and University Courses: The above points for hands-on, real-world projects, and case studies within core Computer Science courses as well as university courses should focus on solidifying the foundational knowledge of the student first. In addition, interdisciplinary courses that combine aspects of core Computer Science knowledge with other fields of study (e.g. business, design, communication, etc.) would allow Gen Z students to experience learning in many different ways and see how the core knowledge of Computer Science can be applied in many different contexts.
2. Support systems: Offer students more hands-on mentorship as well as programs, services and events which promote mental health and well-being through counseling as well as via wellness and other type of workshops and events.
3. Technology-Enhanced Learning: Utilize interactive learning platforms and propose the gamification of learning.
4. Data-Driven Decision Making; continue to regularly assess and gather feedback for the university to use in order to make the best decisions in terms of the academic and co-curricular support and services offered for students to be successful in their academics and in life after CSU. Use of Longitudinal studies [23] to track Computer Science students for multiple years to assess the long-term effects of different course types, teaching methods, as well as academic and co-curricular support systems offered to students in order to make decisions that will promote student success in the academic programs offered by the university.
5. Develop Collaborative Learning Environments: Project-based learning and setting up industry partnerships with local organizations, companies, and employers to facilitate co-op programs, internships, and projects that Gen Z students can work on collectively.

Acknowledgment

The author would like to thank the Deanship of Admission and Registration at Al-Baha University for facilitating this research.

References

[1] Ramesh, Dr. Bridging the Gap: The Pervasive Application of Computer Science in Commerce. Journal of Electronics, Computer Networking and Applied Mathematics. (2023):3:1-7.

[2] National Academies of Sciences, Engineering, and Medicine. 2018. Assessing and Responding to the Growth of Computer Science Undergraduate Enrollments. Washington, DC: The National Academies Press.

[3] Zagami, J. Cognitive Influences on Learning Programming. In: Zagami, J, editor. Teaching Coding in K-12 Schools: Research and Application, Grifith University Press, 2023.

[4] Obaido, G, Agbo, F.J, Alvarado, C and Oyelere, S.S. Analysis of Attrition Studies Within the Computer Sciences, IEEE Access, (2023):11:53736-53748.

[5] Ozkan, F & Kettler, T. Effects of STEM education on the academic success and social-emotional development of gifted students. (2022):9:143-163.

[6] Sofowora, M & Eyono Obono, S & Abayomi, A. The Influence of Mathematics on Students' Performance in Computer Programming. (2022):5:745-755.

[7] Cheng, G. The Effects of Self-regulated Learning Strategies on Computer Programming Achievement in Teacher Education. Frontiers in Education Technology. (2021):4: p37.

[8] Schoeffel, P, Wazlawick, R.S, and Ramos, V.F.C. Motivation and Engagement Factors of Undergraduate Students in Computing: A systematic mapping study, 2018 IEEE Frontiers in Education Conference (FIE), San Jose, CA, USA, 2018:1-5.

[9] Liu J, Sun M, Dong Y, Xu F, Sun X and Zhou Y. The Mediating Effect of Creativity on the Relationship Between Mathematic Achievement and Programming Self-Efficacy. Front. Psychol. (2022):12:772093.

[10] Ishak, N, Yusoff, N & Madihie, A. Resilience in Mathematics, Academic Resilience, or Mathematical Resilience?: An Overview. Universal Journal of Educational Research. (2020):8:34-39.

[11] Varga, E. B., & Sátán, Á. Detecting at-risk students on Computer Science bachelor programs based on pre-enrollment characteristics. Hungarian Educational Research Journal, (2021):1(3):297-310.

[12] Mendoza Diaz, N. V, Trytten, D. A, Meier, R, & Moore, J. M. The Impact of Prior Programming Experience on Computational Thinking in First-Year Engineering Experience. Proceedings of Collaborative Network for Engineering & Computing Diversity Conference, 2022, February , New Orleans, Louisiana, .

[13] Trudy Howles. Preliminary results of a longitudinal study of computer science student trends, behaviors and preferences. J. Comput. Sci. Coll. (2007):22:6:18–27.

[14] A. Dolot, The characteristics of Generation Z. e-mentor (2018):2:44-50.

[15] Jawad, H.M, and Tout, S. Gamifying Computer Science Education for Z Generation," Information, vol. 12, no. 11, p. 453, Nov. 2021.

[16] Beaudoin, L and Avanthey, L. How to help digital-native students to successfully take control of their learning: A return of 8 years of experience on a Computer Science e-learning platform in higher education, Educ. Inf. Technol., (2023):28:5421–5451.

[17] Chan, C.K.Y., Lee, K.K.W. The AI generation gap: Are Gen Z students more interested in adopting generative AI such as ChatGPT in teaching and learning than their Gen X and millennial generation teachers? Smart Learn. Environ. (2023):10:60.

[18] Sayago Barrantes, S. Are Voice Assistants Suitable Learning Companions for Generation-Z students of Computer Science in the Classroom? A Pilot Study, Revista del CIDUI, Proc. Congrés Internacional de Docència Universitària i Innovació (CIDUI).(2021): 5:1-8.

[19] Saaty, T.L. Fundamentals of Decision Making and Priority Theory with the Analytic Hierarchy Process. Pittsburgh, PA: RWS Publications, 1994.

[20] Stofkova, J. et al., Use of the Analytic Hierarchy Process and Selected Methods in the Managerial Decision-Making Process in the Context of Sustainable Development, Sustainability. (2022):14(18):11546.

[21] Wang, N, Ren, Z, Zhang, Z, Fu J. Evaluation and Prediction of Higher Education System Based on AHP-TOPSIS and LSTM Neural Network, Applied Sciences. (2022):12(10): 4987.

[22] Abastante, F, Corrente, S, Greco, S, Ishizaka, A, and Lami, I.M. A new parsimonious AHP methodology: Assigning priorities to many objects by comparing pairwise few reference objects. Expert Syst. Appl. 127, C (Aug 2019):109–120.

[23] Björn, C. A Longitudinal Study Following CS Students' Progression. Proceedings of the 27th ACM Conference on Innovation and Technology in Computer Science Education, New York, NY, USA, (2022):2:642–643.