[1] Omar Ibrahim Asiri

[2] Abdelrahman Osman Elfaki

[3] Mohamed Esmael Abushaira

Introducing the Sign Language Flowchart: Towards Enhancing Programming Learning for Students with Hearing Disabilities

Abstract: - There is a dependence on software applications in many aspects of contemporary life. Teaching programming to individuals has become one of the focal points in the strategic plans of many countries. It is evident that students with hearing disabilities face difficulties in learning programming, which means a significant percentage of society encounters challenges in acquiring programming skills. The first step in learning programming is understanding flowcharts and using them to design simple programs. Therefore, to address the difficulties faced by students with hearing disabilities, we have translated flowcharts into sign language, resulting in the creation of Sign Language Flowcharts. To scientifically explore the performance of students with hearing disabilities in understanding flowcharts, a series of training sessions was conducted. The results confirmed the difficulties faced by students with hearing disabilities in grasping flowchart concepts. We took the previous results as motivation to develop Sign Language Flowcharts, which is the outcome of translating flowcharts into sign language. The results demonstrate the effectiveness of Sign Language Flowcharts in enhancing the understanding of flowchart concepts for students with hearing disabilities.

Keywords: Sign Language, Flowchart, Hearing Disabilities

1. Introduction

Sign languages serve as a communication bridge, connecting individuals with speech impairments to those without, thereby enabling effective interaction. Students with hearing disabilities primarily rely on sign language to understand academic courses, and this, of course, includes computer science students who are deaf. Research has consistently shown that deaf students face challenges in achieving proficiency in reading and writing skills, resulting in lower educational achievement compared to their hearing peers (Salkić & Povlakic, 2022). According to Cavender et al. (2009), deaf and hard-of-hearing students are a marginalized population in the field of computing, and they encounter additional obstacles when it comes to pursuing higher education in computer science. Research has indicated that students who are deaf or have hearing impairments encounter significant challenges when learning applied fields like medicine, engineering, and computer programming (Abuzinadah et al., 2017). According to Angelopoulou et al. (2023), traditional computer science courses primarily concentrate on imparting technical skills, neglecting the essential knowledge and training required for creating accessible and inclusive applications that cater to individuals with hearing disabilities. In addition, Distante and Huang (2007) and Jašková and Stankovičová (2022) claimed that providing an interpreter in sign language was necessary for computer science students with hearing impairments.

On the other hand, one of the primary goals of computer science educators is to promote equity by ensuring that every student has the opportunity to acquire fundamental computer science knowledge. Hence, the previous two points highlight the importance of finding a mechanism to explain the concepts of computer programming courses to students with hearing disabilities, ensuring that they are on par with their fellow regular students.

Prior research encouraged the development of new tools for people with disabilities to assist them in actively participating in educational activities Elfaki and Bassfar (2019), Mohamed et al. (2019), and Elfaki and Alotaibi (2018).

While Elfaki and Bassfar (2020) claimed that writing software code is one of the main activities in computer science education. Furthermore, Alsaadi et al., (2022) claimed that Arabic sign language is one of the most used sign languages. Regrading to this popularity, and for the need of enabling deaf students to learn programming using their communicated language, we have translated the flowchart diagram into Arabic sign language. The new developed flowchart is called the "Sign Language Flowchart". Elfaki & Bassfar (2020) stated that the flowchart is the first step in learning programming.

In this research, we followed a methodology consisting of three steps. During the first step, we conducted experiments to investigate how students with hearing disabilities comprehend the traditional flowchart diagram.

[1] Information Technology Department, College of Computers and Information Technology, University of Tabuk, Saudi Arabia. oasiri@ut.edu.sa
[2] Computer Science Department, College of Computers and Information Technology, University of Tabuk, Saudi Arabia. a.elfaki@ut.edu.sa
[3] Special Education Department College of Education and Arts, University of Tabuk, Saudi Arabia. mabushaira@ut.edu.sa

In the second step, the Sign Language Flowchart was developed and presented. In the third step, the validation of the developed Sign Language Flowchart was introduced.
This paper is organized as follows: in Section Two, the related works have been analyzed, and the research gap has been identified and presented. In Section Three, the details of developing the Sign Language Flowchart have been presented. Finally, a discussion and conclusion have been presented.

2. Related Works

Andrei et al. (2013) developed a signing avatar for teaching Science, Technology, Engineering, and Mathematics (STEM) courses to deaf students. This work addresses the problem of the lack of sign language specialists in STEM courses. Ng'ethe et al. (2015) developed mobile application that provides an authoring tool to construct computer science lessons by using a series of signed language videos and images, hence a deaf person can teach computer literacy skills. Abuzinadah et al. (2017) conducted experiments to demonstrate the essential requirement for a specialized tool to assist deaf individuals in learning computer programming. Thanasekhar et al. (2019) developed a real-time sign language recognition system to detect gestures from the Indian Sign Language (ISL) dictionary using a Convolutional Neural Network model. This system is capable of identifying the words being communicated, including basic programming keywords (Python- like syntax), which have been learned from dataset containing 500 different images of the gesture corresponding to each word.
Ladner et al. (2020) conducted experiments to explore how well teaching CS courses meets the needs of deaf students by using Code.org. Their results ensure the critical need for new tools for teaching deaf students the fundamentals of computer science. Ward et al. (2021) developed a prototype of a centralized portal to provide accessible virtual learning for students with disabilities, specifically focusing on teaching computer science modules. However, this prototype neglected to address the needs of students with hearing disabilities, as it did not include sign language as a means of communication. Levinson et al. (2021) conducted classroom experiments to measure the understanding of coding and computational thinking education among students with hearing disabilities. Their results demonstrate the need for specialized educational tools to improve these student's outcomes. Srikantaswamy (2022) proposed a technique built on a keyword-based concept to teach deaf students computer sciences courses. Then, a visual interpretation of the listed keyword was developed. The derivation of sign language for the keyword is primarily based on concept-oriented principles.
Marghitu et al. (2023) developed coding instruction videos for K12 hearing impaired students using American Sign Language. The aim is to address the shortage of sign language translators who understand computer coding. Usman (2023) investigated the impact of considering hearing disabilities in the design of web educational tools. However, the work did not provide any suggestions for implementing a solution. Aljedaani et al. (2023) examined the impact of online teaching of computer science courses during the COVID-19 pandemic on deaf students. The results demonstrated the significance of providing a better learning experience for deaf and hard-of-hearing students. Angelopoulou et al. (2023) developed experiments to encourage computer science students to use accessibility methods in their applications. The experiments were conducted to incorporate accessible applications into computer science courses, followed by an assessment of the effects of accessibility-related activities and course delivery modes on students' understanding of accessibility. The findings suggest that students exhibited increased confidence, interest, and familiarity with accessible technology because of these initiatives.

3. Developing the Sign Language Flowchart

Motivation

Teaching computer programming to individuals with hearing disabilities can present unique difficulties and challenges. Researchers have conducted various studies, utilizing real-time experiments, interviews, and questionnaires to investigate these challenges (Archundia-Sierra et al.,2023). The same phenomenon has occurred in the specific case of the Diploma of Computer Programming course offered by the Applied College at the University of Tabuk, where instructors have also experienced similar difficulties and have expressed their concerns regarding these challenges. While the exact nature of these challenges may vary, some common issues could include communication barriers, difficulty to understand course materials, and finding alternative course materials that could be effective to the specific needs of students with hearing disabilities. Communication barriers can arise due to the reliance on verbal instructions and explanations in traditional materials of programming education.
Consequently, instructors may need to explore alternative methods to explain the materials of programming education. Teaching methods also need to be adapted to meet the unique learning needs of individuals with hearing disabilities. Instructors may need to employ visual programming tools, demonstration-based teaching, and interactive activities that rely less on auditory cues.
Therefore, in this paper, a flowchart has been selected as a programming design tool that acts towards enhancing computer programming teaching for people with hearing disabilities. A flowchart has been selected, as it is usually the first technique that programming students should learn (Almasabe et al, 2023; Duan et al., 2016; Abdalla et

al., 2019). We have developed a Sign Language Flowchart, which utilizes sign language instead of traditional instructions, to enhance accessibility in our teaching approach.

Study sample:

For the participants, we used a purposive sample taken from the students of the computer programming diploma program for deaf students offered by the Applied College at the University of Tabuk in the academic year 2023-2024. Nineteen students were able to participate and complete the study. To comply with research ethics, the authors obtained permission from the Institutional Review Board (IRB) to conduct the study. The study was conducted during the Spring semester of the 2024 academic year.

The study procedures

The goal of this study was to find out whether using sign language flowcharts would help students to obtain better results in learning programming. Initially, we conducted a pilot test to assess the students' comprehension of flowcharts. The findings revealed that all students faced significant challenges in acquiring a complete and accurate understanding of the presented flowchart. As a result, we decided to establish training sessions specifically geared towards teaching students how to interpret and utilize flowcharts. Subsequently, we administered a quiz to evaluate the students' knowledge of flowcharts. The results clearly indicated the difficulties encountered by the selected students in comprehending flowcharts. As a result of that, we developed the Sign Language Flowchart. Students were then provided another training session on the new Sign Language Flowchart. The session focused on teaching the students the flowchart elements that were translated into sign language. Finally, we assessed the impact of our proposed Sign Language Flowchart on improving students' understanding of flowchart concepts. The results of using our developed Sign Language Flowchart are highly promising. Further details are demonstrated in the results section. Table 1 displays the study procedures and steps from the initial thought to the findings.

Table 1: The study procedures

	Step	Input	Output
1	Pilot test to assess the students' comprehension of the proposed flowchart	Questions, Interviews	Need to provide training sessions about flowcharts
2	Conduct training session to teach students how to interpret and utilize flowcharts	Training materials	Training sessions
3	Assess the results of the training sessions	Test questions	Enhance flowchart knowledge slightly compared to the first step
4	Develop the Sign Language Flowchart	Traditional flowchart, Signs of written Arabic sign language	Sign Language Flowchart
5	Conduct training sessions using the Sign Language Flowchart	Training materials	Training sessions
6	Assess the developed Sign Language Flowchart	Test materials	Report the findings which indicated that the Sign Language Flowchart significantly enhanced students' understanding of flowchart concepts

In the following, each step of our study procedure has been explaied in detail.

1) Assess the students' comprehension of the flowchart. As is known to all programming instructors, flowcharts are the first step in explaining programming. These help students to understand the logic of programming by illustrating the sequential steps involved in its execution. Therefore, when assessing students' comprehension of programming, it is necessary to evaluate first their understanding of flowcharts. This step has been implemented by conducting direct interviews with all the participants in our study. We then distributed questions gathered from flowchart classes. The results indicate a crucial need to improve the understanding of flowcharts to enhance computer programming skills for deaf individuals.

2) Conducting training sessions to teach students how to interpret and utilize flowcharts. The participants were offered multiple sessions related to this matter. Table 2 illustrates the training program schedule.

Table 2: The training program schedule

#	Session Title	Purpose	Training Content	Evaluation Questions	Duration
1	Introductory session	Getting to know the students and introducing the program	The concept of flowcharts, their importance and use	What are flowcharts? What is their importance? What are their uses?	50 Min
2	Algorithms	Defining concepts of the algorithms	Learn about the concept of algorithms Learn about the importance of algorithms Learn about the characteristics of algorithms	What are the steps to make a phone call?	50 Min
3	Structure of alogrithms	Defining structure of alogrithms	Concept of sequencing Concept of choice Concept of repetition	Provide examples of a sequence, repetition, and selection	50 min
4	Exercise	Ensure the understanding of the previous part	Arithmetic problems as content for algorithm exercise	Solving arithmetic problems by using algorithms	50 min
5	Flowchart (1)	Understanding the flowchart shapes (part 1)	Start and end shapes Input and output shapes Processing and operations	Recognize the discussed shapes by using examples	50 min
6	Flowchart (2)	Understanding the flowchart shapes (part 2)	Conditions shapes Repretions shaper	Recongize the discussed shapes by using examples	50 min
7	Exercise	Ensure the understanding of the previous part	Arithmetic problems as content for flowchart exercise	Solving arithmetic problems by using flowcharts	50 min
8	Test	Ensure the understanding of the conducted training			120 min

3) Assess the results of the training sessions. The last step in Table 2 illustrates the test that has been conducted to evaluate the results of the provided training sessions. The results show a slight enhancement of flowchart knowledge compared to the previous findings. However, there is still a severe lack of understanding in flowchart knowledge, which motivated us to develop the Sign Language Flowchart.

4) Develop the Sign Language Flowchart. In this step, we have used the symbols of written Arabic sign language to re-represent the elements of flowcharts in form of sign language. Figure 1 shows an example of written Arabic sign language.

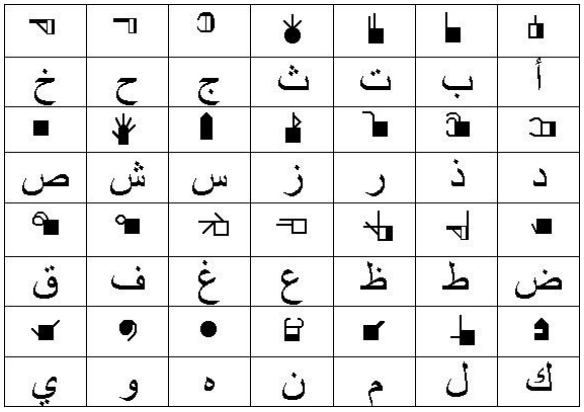


Figure 1: Example of written Arabic sign language

During this stage of the study we translated the flowchart elements into written Arabic sign language to produce the new concept, which is the Sign Language Flowchart. Table 3 illustrates the main translated shapes.

Table 3: The proposed translated shapes of the flowchart

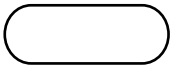
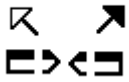
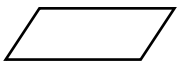
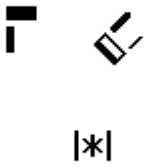
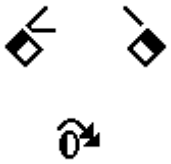
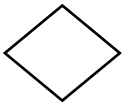
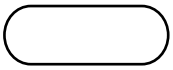
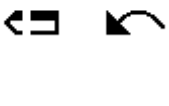
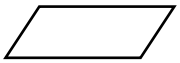
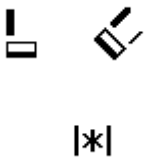
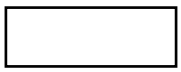
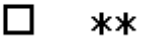
#	Flowchart meaning	Shape in regular flowchart	Element of the written Arabic sign language	Sign Language translation of the flowchart element
1	Start			
2	Input			
3	Processing			
4	Decision			
5	End			
6	Output/print			
7	Process			

Figure 2 is an example of one of the Sign Language Flowcharts used in the study. The problem that needs to be solved in the figure is a sum of two numbers. Thus, two variables A, and B have been entered into the system, and their total has been printed as a result.

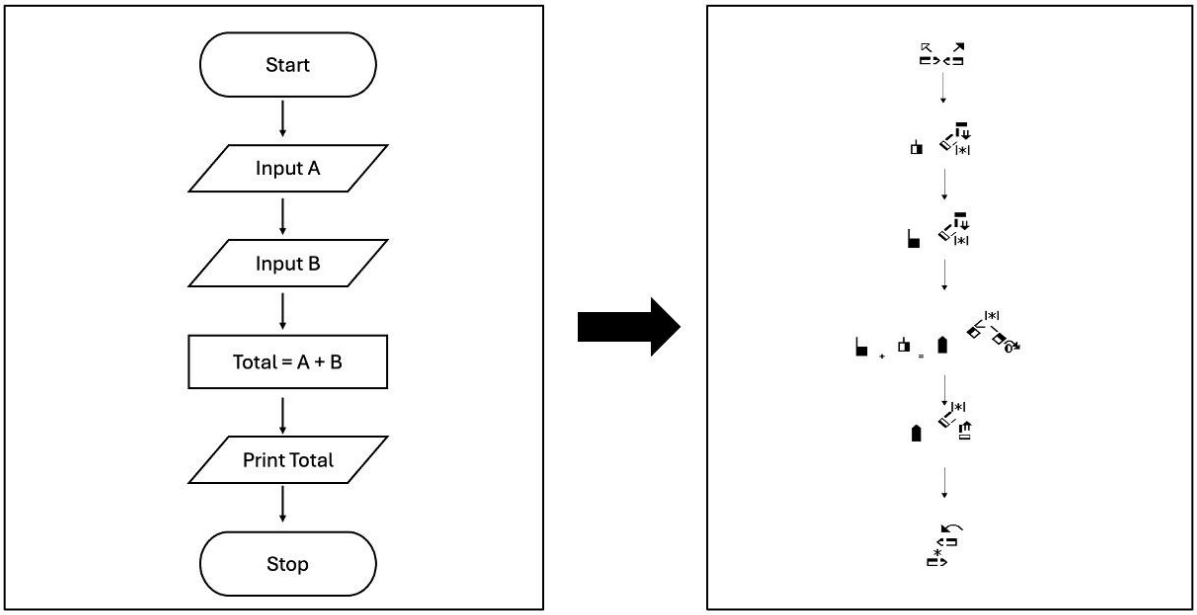


Figure 2: Example of the Sign Language Flowchart

5) In this step, we have taught programming for the sample students using the developed Sign Language Flowchart. Participants received a training session on the Sign Language Flowchart, in which the participants were given the opportunity to practice and apply their knowledge. Figure 3 depicts an image from the training session.

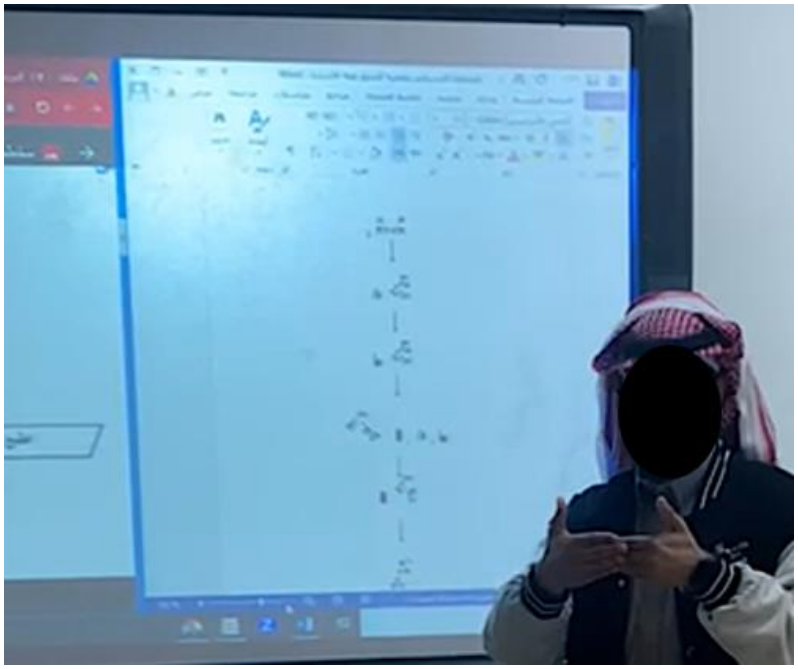


Figure 3: A participant practices the Sign Language Flowchart

6) Assess the developed Sign Language Flowchart. After finishing two training sessions, students were provided with a test consisting of 10 multiple choice questions. The questions were designed to find out if using sign writing would help students to obtain a better understanding of flowchart concepts. Figure 4 depicts the questions used in the test. The results of the test in this step were promising, as the majority of the questions were answered with very high scores. The participants' average on the test was 8.74 out of 10, with a standard deviation of 0.81. Table 3 illustrates the individual score for each participant.

Table 3: individual score for participants

Participant #	Score
1	9
2	7
3	9
4	7
5	9
6	10
7	9
8	8
9	9
10	10
11	9
12	8
13	9
14	9
15	9
16	9
17	9
18	8
19	9
Average	8.74
SD	0.81

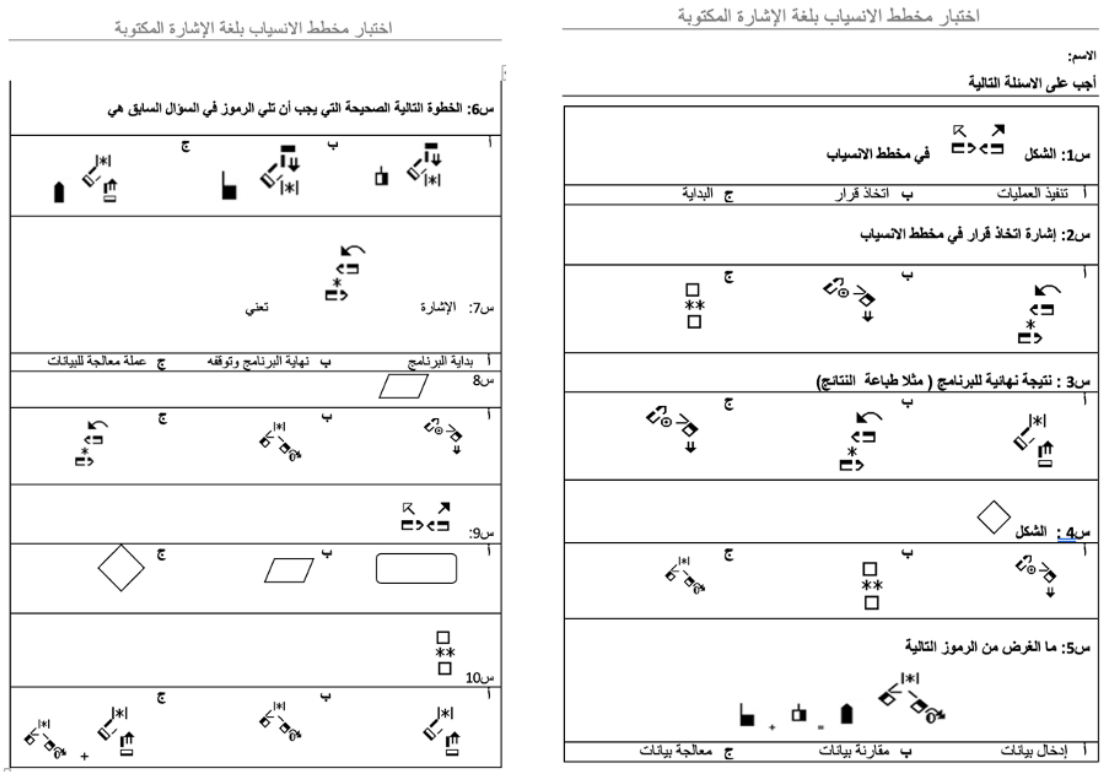


Figure 4: Sample of questions used in the test

The test results reflected promising findings, indicating that the effectiveness of the developed Sign Language Flowchart. Most of the participants were able to achieve high scores. In terms of aggregated results for each question, 74% of the questions scored above the average. The aggregated findings for each question indicate that 70% of the posttest questions had at least 89% accurate responses. Table 4 and Figure 5 illustrate the detailed findings regarding the answers' accuracy.

Table 4: Findings about answer accuracy for each question

	Questions									
Question number	Q 1	Q 2	Q 3	Q 4	Q 5	Q 6	Q 7	Q 8	Q 9	Q 10
Correct answers	19	19	13	18	17	17	14	12	18	19
% of correct answers	100%	100%	68%	95%	89%	89%	74%	63%	95%	100%
Mean	0	0	0.48	0.23	0.32	0.32	0.45	0.50	0.23	0
SD	1	1	0.68	0.95	0.89	0.89	0.74	0.63	0.95	1

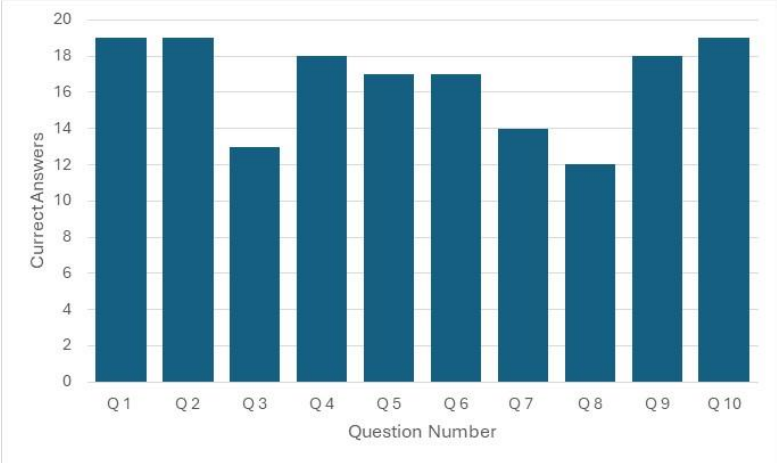


Figure 5: Representation of the participants' results on each question

4. Conclusion and Discussion

Scientific studies have proven that individuals with hearing disabilities struggle to communicate effectively using traditional methods. The most successful methods of communication for them rely on sign language (Asiri et al., 2024). Therefore, it is crucial to introduce a method to transfer educational tools into sign languages. This research deals with the issue of difficulties of teaching programming for students with hearing disabilities. In this research, the flowchart was chosen to be translated into sign language, as it is considered the first step students take in their journey to learn programming.

Initially, the knowledge of deaf students about flowcharts was assessed through group quizzes and individual interviews. The results confirmed the need to provide training sessions on flowcharts. Secondly, training sessions were conducted to teach students with hearing disabilities the principles of basic programming using traditional flowcharts. Third, the difficulties students faced in understanding were authenticated and proven through evaluation tests. Fourth, these proven difficulties motivated us to incorporate sign language into the programming learning process. We used written Arabic sign language to translate traditional flowcharts, and we refer to the translated flowchart as the Sign Language Flowchart. Training sessions were conducted using the developed Sign Language Flowchart to help deaf students understand the principles of programming. In these training sessions, we used the same training materials that were used in previous sessions, but with a replacement of the traditional flowchart with the Sign Language Flowchart. To evaluate our developed Sign Language Flowchart, we conducted a study where the results showed a significant enhancement in students' understanding of flowchart concepts. Comparing the level of understanding of the concepts by the Sign Language Flowchart, the students achieved a peak result with an average of 87.4%. The overall result for most questions was around 90%.

The revealed results are aligned with what Salkić and Povlakic (2022) indicated that hard-of-hearing students are not proficient in reading. Thus, transferring knowledge from regular written languages into sign language would help them to have a better understanding of the information. As a part of the validation for this hypothesis, results have verified these findings.

For future work, additional tools for computer science learning and education will be translated into sign language, with the aim of enhancing the educational process of programming for deaf and hard-of-hearing students.

Acknowledgment
The authors extend their appreciation to the Deanship of Research and Graduate Studies at University of Tabuk for funding this work through Research no. S-1444-0087.

References

[1] Abdalla, O. A., Aborokbah, M. M., & ELfaki, A. O. (2019). Developing classification model to investigate the problem of computing students studies length. International Journal of Machine Learning and Computing, 9(2), 174-180.
[2] Abuzinadah, N. E., Malibari, A. A., & Krause, P. (2017). Towards Empowering Hearing Impaired Students' Skills in Computing and Technology. International Journal of Advanced Computer Science and Applications, 8(1).
[3] Aljedaani, W., Aljedaani, M., Mkaouer, M. W., & Ludi, S. (2023, February). Teachers Perspectives on Transition to Online Teaching Deaf and Hard-of-Hearing Students during the COVID-19 Pandemic: A Case Study. In Proceedings of the 16th Innovations in Software Engineering Conference (pp. 1-10).
[4] Almasabe, A., Ludi, S., & Elfaki, A. O. (2023). Scrutinizing UML Teaching and Learning Modeling Tools.
[5] Andrei, S., Osborne, L., & Smith, Z. (2013). Designing an American Sign Language avatar for learning computer science concepts for deaf or hard-of-hearing students and deaf interpreters. Journal of Educational Multimedia and Hypermedia, 22(3), 229-242.
[6] Angelopoulou, A., Hodhod, R., Lilly, K., & Newland, A. (2023). Introducing Accessible Design to Students in Computer Science. Journal of Postsecondary Education & Disability, 36(1).
[7] Alsaadi, Z., Alshamani, E., Alrehaili, M., Alrashdi, A. A. D., Albelwi, S., & Elfaki, A. O. (2022). A real time Arabic sign language alphabets (ArSLA) recognition model using deep learning architecture. Computers, 11(5), 78.
[8] Archundia-Sierra, E. (2023). Design of a Tool to Promote Teacher Training in the Inclusive Classroom in Care for Children with Hearing Disabilities Mediated by ICT. Programming and Computer Software, 49(8), 790-795.
[9] Asiri, O. I., Elfaki, A. O., & Abushaira, M. E. (2024). Investigating The Use Of Leap Motion Controller In Recognition Of Arabic Sign Language. Educational Administration: Theory and Practice, 30(5), 1245-1254.
[10] Cavender, A. C., Ladner, R. E., & Roth, R. I. (2009). The summer academy for advancing deaf and hard of hearing in computing. ACM SIGCSE Bulletin, 41(1), 514-518.
[11] Distante, D., & Huang, S. (2007, July). Challenges and lessons learned in teaching software engineering and programming to hearing-impaired students. In 20th Conference on Software Engineering Education & Training (CSEET'07) (pp. 344-354). IEEE.
[12] Duan, Y., Narendra, N. C., Gao, H., Tang, M., Elfaki, A. O., Wan, S., & Lu, J. (2016). Variability based Approach for Minimizing over Design and under Design. In MATEC Web of Conferences (Vol. 54, p. 07002). EDP Sciences.
[13] Elfaki, A. O., & Alotaibi, M. (2018). The role of M-health applications in the fight against Alzheimer's: current and future directions. Mhealth, 4.
[14] Elfaki, A. O., & Bassfar, Z. A. I. D. (2019). Auto-measuring usability method based on runtime user's behavior: case Study for governmental web-based information systems. Journal of Theoretical and Applied Information Technology, 97(13).
[15] Elfaki, A., & Bassfar, Z. (2020). Construction of a software development model for managing final year projects in information technology programmes. International Journal of Emerging Technologies in Learning (iJET), 15(21), 4-23.
[16] Jašková, Ľ., & Stankovičová, M. (2022). Teaching Computer Science Students with Special Needs During Distance Education. In INTED2022 Proceedings (pp. 2068-2075). IATED.
[17] Ladner, R. E., Stefik, A., Naumann, J., & Peach, E. (2020, March). Computer science principles for teachers of deaf students. In 2020 Research on Equity and Sustained Participation in Engineering, Computing, and Technology (RESPECT) (Vol. 1, pp. 1-4). IEEE.
[18] Levinson, T., Hunt, L., & Hassenfeld, Z. R. (2021). Including students with disabilities in the coding classroom. In Teaching computational thinking and coding to young children (pp. 236-248). IGI Global.
[19] Marghitu, D., Howard, A., & Das, M. (2023). Developing Coding Instruction Videos for K12 Hearing Impaired Students Using American Sign Language. ACM SIGCSE Bulletin, 55(2), 10-12.
[20] Mohamed, M. H., Elfaki, A. O., Johar, M. G. M., Younoussi, S., el Rhaffari, I., Amoud, M., ... & Denggao, J. Z. (2019). On Auto-measuring of Applications Usability for Blind People. J. Softw., 14(4), 146-152.
[21] Ng'ethe, G. G., Blake, E. H., & Glaser, M. (2015, May). SignSupport: A mobile aid for deaf people learning computer literacy skills. In International Conference on Computer Supported Education (Vol. 2, pp. 501-511). SCITEPRESS.
[22] Salkić, N., & Povlakić Hadžiefendić, M. (2022). READING, WRITING AND COMPREHENSION IN DEAF CHILDREN OF THE ADVERBIAL CLAUSE OF PLACE IN WRITTEN FORM. KNOWLEDGE - International Journal, 54(4), 719–724. https://doi.org/10.35120/kij5404719s
[23] Srikantaswamy, D. (2022). Teaching technique to teach students with hearing impairment using keyword vocabulary method. International Journal of Research in Education and Psychology, 8(2), 20-29.
[24] Thanasekhar, B., Kumar, G. D., Akshay, V., & Ashfaaq, A. M. (2019, December). Real Time Conversion of Sign Language using Deep Learning for Programming Basics. In 2019 11th International Conference on Advanced Computing (ICoAC) (pp. 1-6). IEEE.
[25] Usman, R. (2023). Decoding designers: The work of responsibility and innovation for accessibility. Anthropology of Work Review, 44(2), 57-68.
[26] Ward, D., Hall, Q., & Marghitu, D. (2021, October). An Accessible Portal to Teach Computer Science Modules to Typical and Special Needs Children: A Prototype. In CENTRIC the International Conference on Advances in Humanoriented and Personalized Mechanisms Technologies and Services.