ASEE 2022 ANNUAL CONFERENCE
Excellence Through Diversity
MINNEAPOLIS, MINNESOTA, JUNE 26TH-29TH, 2022 ASEE

Paper ID #36465

Challenges and Experiences of Converting an Assembly Language and Computer Organization Course into an Online Course

Weiying Zhu (Full Professor)

Dr. Zhu is now a Professor of Computer Science and currently serving as the Associate Department Chair in the Department of Computer Sciences at Metropolitan State University of Denver. She received her Ph.D. at Old Dominion University, VA in August 2006. She was a software engineer at Bell Labs, Lucent Technologies from 1999 to 2003. She joined MSU Denver in August 2011. She had been an Assistant Professor in Hampton University, VA from 2006 to 2011.

Challenges and Experiences of Converting an Assembly Language and Computer Organization Course into an Online Course

ABSTRACT

Teaching some courses online in non-online academic programs at traditional higher education institutions not only provides students more flexibility, but also is a possible solution to the increasing student enrollment when space or faculty is a constraint. However, how to address the challenges in online computer science and engineering (CSE) courses and whether student performance is compromised are two critical questions to be answered while considering offering online CSE courses. Our institution is a large public Hispanic-Serving Institution. Our computer science major program is ABET accredited. Our students have very diverse backgrounds and a wide range of learning capabilities and most of them are first-generation college students. Hopefully, our findings may provide useful information to institutions with similar settings.

This paper presents the challenges and experiences of converting a face-to-face Assembly Language and Computer Organization course, required for both computer science and computer engineering majors, into an online course in Spring 2021. In this paper, we first discuss how various course technologies were adopted and how various strategies were developed and used to support online teaching and learning.

To investigate whether the delivery mode change affects student performance, two face-to-face sections of this course, taught most recently by the same instructor, are used for performance comparison in this paper. A t-test is performed to compare online students' cumulative GPAs with face-to-face students' cumulative GPAs to assure neither group has an unfair advantage. Next, t-tests are used to determine whether there is any significant difference in homework scores and exam scores between online students and face-to-face students. Additionally, the box-and-whisker plots are used to graphically describe and compare online students' homework scores, exam scores, and earned term scores with that of face-to-face students. An anonymous survey was designed to get student perceptions of their learning experiences in this online course. The survey analysis and results are reported in this paper. Three questions are about possible negative impact on students' availability during the class time or on their time commitment, or about possible technical difficulty for taking lessons online. Followed are 15 Likert-scale questions on the use of tools, instructional materials, and hands-on learning activities. We found that 1) there is no statistically significant difference between the student performance in the online section and that in the face-to-face sections, 2) among three surveyed challenges, students' time commitment to this online course was most negatively impacted by pandemic-related situations, and 3) student feedbacks on their learning experiences in this online course are generally positive or very positive.

KEYWORDS

online teaching, course technology, assembly language, computer organization, student performance, student survey, t-test

1 Introduction

Computer organization, computer architecture, and assembly language programming cover fundamental contents that are needed in both computer science and computer engineering curricula. Some of the contents in these areas are also taught in electrical engineering courses. Due to the advances in technology and the flexibility of online courses, there has been an increase of online courses offered by traditional higher education institutions [1], [2]. Our institution is a large public Hispanic-Serving Institution. Our B.S. in Computer Science major program is ABET accredited. We have a very diverse student body in terms of student backgrounds and learning capabilities. The Assembly Language and Computer Organization course is a core course required in both Computer Science (CS) major and Computer Engineering (CpE) major programs at our institution and had always been taught face-to-face before the pandemic. In 2020-21, we converted this course into an online course.

The study presented in this paper aims to summarize our experience of converting an assembly language and computer organization course from the traditional face-to-face mode to the online, to investigate whether the change of the delivery mode affected student performance, and to explore student perceptions of their learning experience and student opinions on the instructional materials and learning activities in the online mode. This paper presents, in the online mode, how course technologies were used for teaching and learning, how the µVision IDE virtual app was used to support hands-on ARM assembly language programming, how learner interactions were supported, and how the challenges in assessment and measurement were addressed. To evaluate the effectiveness of online learning and teaching, a t-test was first used to compare average cumulative GPAs of online students with that of face-to-face students to assure that there is no student background advantage in either the online sample or the face-to-face sample. Next, t-tests were performed between the homework and exam scores of online students and that of face-to-face students to study whether there is any statistically significant difference. Box-and-whisker plots are also used to graphically describe and compare the homework scores, exam scores, and earned term scores of online students and that of face-to-face students. An anonymous student survey was developed and given to online students to survey the negative impacts that may have been caused by the pandemic to their learning and to investigate their opinions on the instructional materials and tools used in this online course. The data and analysis results reported in this paper indicate that 1) there is *no* statistically significant performance difference between online and face-to-face students, 2) among three surveyed challenges, *students' time commitment* was the one most negatively impacted by the pandemic or related situations, and 3) student feedbacks on teaching and learning in the online mode are generally *positive* or *very positive*.

The rest of this paper is organized as follows. Related work is introduced in Section 2. Our assembly language and computer organization course, our online delivery mode, and the goals of our study are described in Section 3. Section 4 presents our experience of converting this course from the traditional face-to-face mode into the online mode. Section 5 evaluates the effectiveness of online learning and teaching through comparing the student performance in an online section and the student performance in two face-to-face sections and analyzing the data of an anonymous student survey given in this online course. Finally, our study is concluded, and the future work is discussed in Section 6.

2 Related work

Topics on computer organization, computer architecture, and/or assembly languages are taught in various computer science (CS), computer engineering (CpE), or electrical engineering (EE) courses at many institutions. In [3], Erdil et al present their design of a lower division computer organization and architecture course with hands-on components. They designed blended-learning modules to introduce core computer design concepts, and hands-on activities and problem-based modules with the flexibility to be applied in face-to-face, synchronous online, or asynchronous online courses. In [4], Luo discusses a back-and-forth based pedagogy integrated with the student-centered learning in Computer Architecture course. Luo summarizes the topics covered in this course and evaluates this pedagogy via student self-assessments. In [5], Peterson et al introduce a free visual simulation of a very simple 4-bit architecture computer that can be used in a digital logic course or a computer organization/architecture course. In [6], Edekopp et al discuss the results of their study of a "flipped" classroom pedagogical approach in their computer architecture course and detail the strategies and tools that they used for in- and out-of-class activities. They studied the effectiveness of the flipped classroom components by analyzing student survey results and compared the student performance by showing average student scores in the "flipped-classroom" approach vs. in the traditional approach. In [7], Yildiz et al share the details (course curriculum, student body, projects, and components) of a project-based microcontroller course and presents feedbacks freely written by students.

Educational researchers have been exploring and studying the online teaching mode in CS, CpE, and EE courses. In a Database course [8], the t-tests indicate that face-to-face students significantly outperformed online students when there were no significant differences in background characteristics. Whitney et al explored the impact of captions on learning performance in an online Intro to Computers and Info Processing course [9]. By dividing 25 students into two subgroups to watch videos with or without captions, they found a performance improvement due to watching videos with captions. Motogna et al studied the change of student learning assessment caused by the sudden transition to online in Software Engineering courses [10]. The analysis on 29 survey responses revealed a trend of having fewer exams and more project-related evaluation and an increase of instructors' effort in learning assessment although not all instructors changed their evaluation methods. In [11], Benson et al describe their experience with converting a face-to-face Intro to Digital Design course online and compare the performance of the students in an online semester and that of the students in a face-to-face semester. Their data indicate that online students performed at least as well as face-to-face students. In [12], Wang et al discuss the conversion of their face-to-face Digital Design Fundamentals course into an online format. Their comparison between the average student scores on a set of learning outcomes in online vs. face-to-face sections shows that online students performed better on some outcomes while face-to-face students performed better on the others.

3 The course, the online delivery mode, and the goals of this study

Our assembly language and computer organization course is a lower-division course, in which students study an assembly language, the development of assembly language programs in a contemporary development environment, error detecting and correcting codes, the functional organization of computers, multiprocessing, and high-performance storage. In the course sections studied in this paper, the ARM assembly language is taught and μVision IDE[1] is used as the development environment for ARM assembly language programming.

In Spring 2021, we converted this course into a synchronous online course. Lecture videos were pre-recorded for most class periods and the rest were used for meetings, labs, or exams. Students were given the flexibility of watching lecture videos during or before class time.

One *goal* of this study is to summarize and discuss our experience of converting this assembly language and computer organization course from the traditional face-to-face mode to the online mode as well as the challenges we addressed. Secondly, this study *aims to* investigate whether changing the delivery mode affects student performance in this course. If there is any, what is the difference. The third *goal* is to explore student perceptions of their learning experience and student opinions on the instructional materials and learning activities in the online mode, given that CS courses were never taught online at our institution before the pandemic.

4 Converting the assembly language and computer organization course online

While converting the assembly language and computer organization course into an online course, we referred to the instructional design in its traditional face-to-face version and made sure to implement all course components in the online format. Most importantly, the course contents and the learning objectives of the online section are the same as that of the face-to-face sections used for performance comparison.

4.1 Course technology for online teaching and learning

Canvas[2] is the primary learning management system (LMS) used in this online course. To support online course delivery, MS Teams[3] and MS Stream[4] were also used. These are three new tools provided and supported by our institution to specifically support online teaching and learning. Additionally, we continued using μVision IDE, Citrix sever and workspace[5], and GlobalProtect[6] in the online mode, all of which were used in the face-to-face (F2F) sections of this course.

All these LMS and tools are provided and maintained by the information technology services (ITS) at our institution. Table 1 describes how the LMS and tools were used to support the teaching and learning in this online assembly language and computer organization course in Spring 2021 at our institution. Canvas and MS Stream were mainly used as the platforms to post instructional materials and assess student learning. Every weekly module in Canvas was made available to students in the middle of the previous week to give students some flexibility on when to watch the pre-recorded lecture videos if they cannot make to the class time. MS Teams was mainly used to hold virtual meetings during class time and online office hours. μVision IDE, Citrix server & workspace, and GlobalProtect were jointly used to support hands-on ARM assembly language programming activities.

[1] https://www2.keil.com/mdk5/uvision/
[2] https://www.instructure.com/canvas
[3] https://www.microsoft.com/en-us/microsoft-teams/group-chat-software
[4] https://www.microsoft.com/en-us/microsoft-365/microsoft-stream
[5] https://www.citrix.com/
[6] https://www.paloaltonetworks.com/products/globalprotect

Table 1: The use of LMS or tools in this online course in Spring 2021

LMS/tools	**How the LMS or tool is used to support online teaching and learning**	**Used in the F2F mode?**
Canvas	1. The *syllabus* section was used to post the syllabus of this course and list all course assignments and online tests in a summary table. 2. The *announcements* section was used to post course announcements. 3. *Pages* were used to post course introduction and policies, institutional policies and resources, ARM and µVision IDE documents, example source code of ARMv7-M/ARMv8-M assembly programs, and PPT slides. These pages were organized into four *modules*. 4. *Pages* were used to post the links to pre-recorded lecture videos and the in-class notes for every class period. These pages were organized into 15 weekly learning *modules*. 5. *Assignments* were used to post homework assignments, collect student submissions, and post grades and graded student work. 6. *Question banks* and *graded quizzes* were used to create three online tests (Exam 1, Exam 2, and Final Exam), collect student submissions, and post grades. 7. *No-submission assignments* were used to post mid-term estimations and final term grades for every student.	No
MS Teams	1. Live *meetings* were scheduled and hosted for some class periods. 2. The *General channel* was used by students to post questions and by the instructor to answer questions post announcements. 3. The *CHAT* app was used for instructor-student video/voice calls during the instructor's office hours.	No
MS Stream	It was used to host pre-recorded lecture videos and the recordings of live meetings during some class periods.	No
µVision IDE	µVision IDE, a window-based professional development kit, was used as the integrated development environment for ARMv7-M/ARMv8-M assembly language programming.	Yes
Citrix server & workspace	The Citrix server hosts the µVision IDE virtual app for Mac users, who used Citrix workspace to access and use this virtual app.	Yes
GlobalProtect	GlobalProtect was used by Mac users to first connect to the institutional VPN from outside campus and then connect to the Citrix server.	Yes

4.2 The use of the µVision IDE virtual app for online assembly programming

In this course, ARM assembly language programming not only is an important topic but also supports several other topics, such as ARM instructions and instruction set architecture, error detecting/correcting codes, hardwired and microprogrammed implementations of the control unit, instruction-level parallelism and processor pipelining, branch prediction, vector processing, and short-vector processing. Even though the syllabus of this course does not specify a particular assembly language or instruction set architecture (ISA) but leaves it to be a choice of the instructor, we believe that ARM is a great choice. First, ARM processors[7] have been widely used in smart phones, tablets, eReaders, digital televisions, servers and networking equipment, automotive embedded systems, microcontrollers, etc. Second, compared to CISC architectures, the RISC-based design of ARM reduces transistor count and, therefore, requires lower costs and less power use. It is interesting to discuss how ARM instructions are designed to achieve such reductions. Thirdly, the design of ARM is closely related to some of the modern computer organization techniques taught in our course. For example, the Advanced SIMD (Single Instruction Multiple Data) [13], an extension to ARMv7 and ARMv8 architectures, is a case of *vector processing*. In this course, we teach ARM's Unified Assembler Language (UAL), which is the common syntax for all modern ARM architectures (v6 and later) [13].

Several years ago, while looking for an ARM simulator to replace the previous one, APM, that supports the outdated ARMv4 architecture, we identified ARMSIM#[8] and µVision IDE to be on our short list for further investigation. Table 2 compares µVision IDE and ARMSIM#. Through investigation and comparison, we eventually chose µVision as the replacement. Firstly, we expect µVision to be kept up to date for new Windows OS and ARM ISA. Being out of date is our biggest concern on the tools developed by academic research teams, such as APM and ARMSIM#. In Summer 2016, we found that ARMSIM# cannot be installed on a couple of Windows 10 computers due to a .net library error. Secondly, although the environment of µVision is more complex than that of ARMSIM#, it is more powerful and has a better usability after the learning curve. Thirdly, using a professional tool that is commonly used in the industry can better prepare our students for their future careers.

Table 2. µVision vs. ARMSIM#

µVision	**ARMSIM#**
It is a professional tool released by ARM Ltd.	It is a tool developed by a research group at University of Victoria.
It is free for academic use if the code size is no more than 32KB.	It is free for academic use.
It is a project manager, editor, and debugger for Cortex-M, Cortex-R4, ARM7, ARM8, and ARM9 processor-based devices.	It is an editor and debugger only for ARM7 processor-based systems.
It supports the UAL syntax.	It supports the GNU syntax.
It *simulates* an ARM core for debugging or *accesses* a Cortex-M based device for real-time debugging, tracing, and analysis.	It *simulates* an ARM7TDMI processor-based system for debugging.
It runs on a Windows operating system (OS).	It runs on a Windows OS or a Linux OS.

[7] http://www.arm.com/products/processors/index.php
[8] https://webhome.cs.uvic.ca/~nigelh/ARMSim-V2.1/index.html

One lab and three programming homework assignments are included in this course for students to study and practice ARMv7-M/ARMv8-M assembly language programming in µVision. Additionally, during the class time when assembly language programing related topics are discussed, live demo is often included to illustrate how to edit, debug, and test ARM assembly language programs in µVision. In Exam 2, students are required to develop and debug a complete ARM assembly language program in µVision. To support these learning activities, µVision has been installed in all Windows computers in all on-campus general computing labs. The instructor used to reserve a Windows general computing lab for the lab and Exam 2 and use a virtual Windows machine for in-class demos. All students may use Windows computers in general computing labs to work on programming assignments. Students who use Windows laptops or workstations at home may also install µVision on their own computers.

A few years ago, to better serve students who use Mac laptops and desktops and to better utilize Mac computers in classrooms and on-campus general computing labs, the ITS at our institution set up and has been providing the µVision virtual app as a cloud-based service inside the institutional private cloud for our assembly language and computer organization course. Figure 1 illustrates the configuration and use of this service. The Citrix server inside the institutional private cloud hosts a variety of Windows-based virtual apps, one of which is the µVision virtual app for our course. The Citrix Workspace, a client program, is used on a Mac laptop or workstation to connect to and work with the Citrix server and to allow Mac users to remotely edit, debug, and test their ARM assembly language programs in the µVision virtual app. On a Mac computer that is outside the on-campus networks, the GlobalProtect Client needs to be executed to connect this Mac computer to the institutional Virtual Private Networks (VPN) first before connecting the Citrix workspace on this computer to the Citrix server. This solution enabled a seamless transition of this course from the face-to-face mode to the online mode in terms supporting all hands-on ARM assembly language programming activities such as in-class demos, the lab, Exam 2, and programming assignments, especially for Mac users. Since the instructor and students in the online course cannot access any on-campus general computing lab, using the µVision IDE virtual app became the only solution for Mac users.

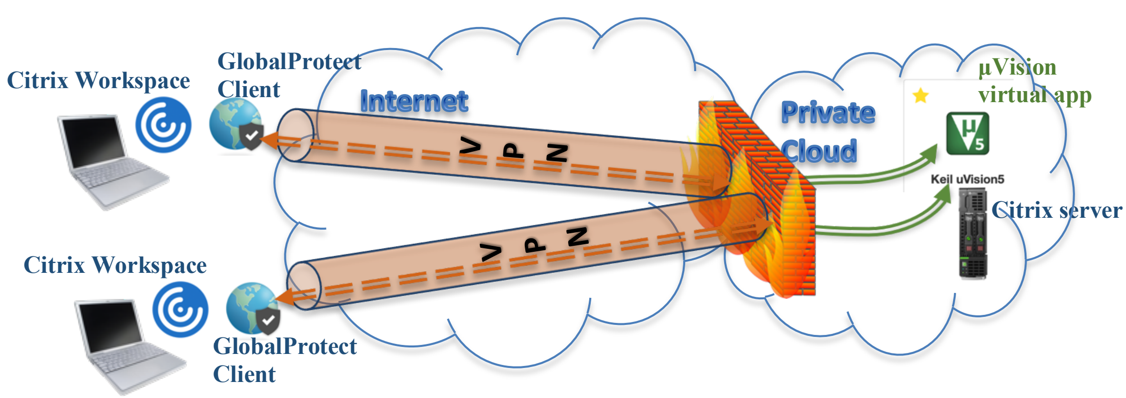

Figure 1. The µVision IDE virtual app solution.

4.3 The support to student-instructor interactions in the online mode

Online learning is different from the traditional face-to-face learning in many ways. Therefore, a

50-minute virtual meeting was held in MS Teams on the first day for overview and introduction. This meeting focused on specific info for online learning, such as where to find instructional materials in Canvas, the schedule of virtual meetings for Labs or Exam Q&As and where to find meeting links, how and where to reach the instructor inside and outside the class time and office hours, and the info on online tests in Canvas.

Supporting teaching and learning activities and student-instructor interactions, both during and outside the class time, is one major challenge in the online mode. To address this challenge, the new approaches that the instructor developed specially for online teaching are 1) pre-recording lecture videos for most of the class periods, uploading lecture videos to MS Stream, and posting the video links in Canvas, 2) pre-recording demo videos on how to edit, debug, and test ARM assembly language programs in the µVision IDE and how to use GlobalProtect and Citrix Workspace, 3) hosting scheduled or on-demand Q&A virtual meetings in MS Teams during class time and making on-demand CHAT video calls during office hours, 4) encouraging students to email questions to the instructor outside class time and office hours and replying student emails promptly, and 5) grading paper-based homework electronically and posting every student's graded work with details of grading and corrections in Canvas. The instructor also continued using most of the approaches used in face-to-face semesters to support hands-on learning activities, such as 1) requesting ITS to set up the µVision IDE virtual app in the institutional private cloud, 2) requesting GlobalProtect accounts for students, and 3) testing students' ARM assembly language programs in the µVision IDE virtual app and posting grading comments with details in Canvas.

4.4 Online student learning assessment

Student learning assessment, especially the assessment part that used to count on exams, is another major challenge that needs to be addressed in the online mode. Besides being part of the learning activities, homework assignments were also used to assess student learning. A homework assignment was created as an "Assignment" in Canvas, with a submission type of "Online" and an online entry option of "File Uploads", for students to submit the source code of their ARM assembly programs or an electronic copy of their paper-based homework.

Exams are a critical component in assessing and measuring student learning outcomes. In this online course, the instructor made all exams into online tests, i.e., "Quizzes" in Canvas. The length of an online test was the same as that of an exam in face-to-face semesters. The question types used in online tests are "Essay Question", "Fill in Multiple Blanks", "Multiple Choice", and "True/False". The instructor developed several strategies to encourage academic integrity in online tests. First, most of the points in an online test were for problems that require students to analyze and solve using the specific info given in the problem statement. Students were required to include sufficient details in their solutions. Less than 10% of the points were for problems on concepts and facts. Secondly, a "Question Bank" consisting of multiple questions was created in Canvas for each problem in an online test and one of those questions was randomly picked for a student when this student moved to a problem while taking the online test. This significantly reduced the chance of any two students getting the same set of questions. Thirdly, the questions in an online test were shown one at a time and a question was locked when a student moved to the next one. Fourthly, all students in class were required to take an online test simultaneously.

Fifthly, the instructor re-sent the course policies on academic integrity with details that particularly apply to this course and reminded students via several ways before every online test. Lastly, when anything suspicious was observed in the submitted work, the “View Log” function under “Moderate This Quiz” in Canvas was used to check student actions during an online test.

As an example, four versions of a problem in an online test for this online course is shown in Figure 2, which partially shows how some of the strategies discussed above were implemented in our design of every online test. By including different ARM addition instructions (circled in a blue circle in Figure 2), we made sure that the solutions and answers to this problem in each version are significantly different from that in the other three versions. These four versions were added into the question bank created for Problem B. During this online test, one of these four versions was randomly picked by Canvas for a student to solve. Meanwhile, students were required to submit their details of work for this problem for the instructor to check. Moreover, students were not able to directly find the answers and solutions to such kind of problems in teaching materials or on the Internet.

Problem B On an ARM processor, **initially**, assuming that [R1] = 0x5D2A8274, [R2] = 0x86AE79BD, [R3] = 0xE0000009, [R4] = 0x0000000C, [N-bit] = 1, [Z-bit] = 1, [C-bit] = 1, [V-bit] = 0, PREDICT [R1] (32-bit value in UPPER-CASE HEX), [N-bit], [Z-bit], [C-bit], and [V-bit] (**0** or **1** for each of these conditional flags) AFTER the following instruction is executed.

(**Requirement**: Keep your details of work for this question and label it clearly with this question number either on your *scratch paper* or in your *e-document*, which will be required to be uploaded altogether in the last question.)

ADCS R1, R1, R3, ROR #4

[R1] = 0x[R1]

[N] = [N]

[Z] = [Z]

[C] = [C]

[V] = [V]

(a) version 1 of Problem B, where the ARM instruction is different from that in three other versions

ADCS R1, R2, R3, ROR #4	**ADDS R1, R3, R4, RRX**	**ADDS R1, R1, R4, RRX**
(b) ARM instruction in version 2	(c) ARM instruction in version 3	(d) ARM instruction in version 4

Figure 2. Four versions of Problem B in an Online Test for this Online Course

5 Evaluating the effectiveness of online teaching and learning

Several quantitative methods were used in this study to evaluate the effectiveness of teaching and learning in this online assembly language and computer organization course. T-tests were used to compare students’ background demographics in terms of cumulative GPAs and then to compare student homework and exam scores of online students and that of face-to-face students. Student homework, exam, and earned term scores were also graphically described and compared via box-and-whisker plots. An anonymous student survey was given in Canvas to study students' perceptions on teaching and learning in this online course.

5.1 Performance comparison between online students and face-to-face students

In Spring 2021, the instructor taught one section of this online assembly language and computer organization course. To investigate whether the change of the delivery mode affects student performance, two face-to-face sections of this course are used for performance comparison in this paper. We decided to use these two face-to-face sections because 1) they were taught by the *same instructor* during *most recent semesters*, 2) using two face-to-face sections makes the size of the *face-to-face sample* (consisting of one section in Fall 2018 and one section in Spring 2018) close to and larger than the size of the *online sample* (consisting of one section in Spring 2021), where the face-to-face sample includes 44 students and the online sample includes 30 students, and 3) using two face-to-face sections makes the cumulative GPA mean of the face-to-face sample and that of the online sample very close to each other. The student learning objectives and course contents in these two face-to-face sections are the same as that in this online course taught in Spring 2021. The section taught by the same instructor in Spring 2020 is not used due to the unclear impact caused by the sudden switch from face to face to online in the middle of the semester.

Based on the sample sizes, one analysis technique we chose is the student's t-test [14] to determine whether there is any statistically significant difference in terms of homework scores and exam scores, respectively, between the online sample and the face-to-face sample. The other analysis technique we chose to use is the box-and-whisker plot [15]. It is used to graphically describe students' scores in the face-to-face or online sample and to visually compare the degree of dispersion and skewness as well as various values (e.g., means, medians, upper and lower quartiles, and ranges) between the online sample and the face-to-face sample.

Table 3 shows the percentages of students' majors in the online v.s. face-to-face samples. While our computer science major program has been in existence for a long time and is ABET accredited, the computer engineering major program at our institution is new and hasn't been ABET accredited yet. This assembly language and computer organization course is a required core course in both major programs. We can see that most students in each sample are majored in CS while the percentage of CpE majors increased over time. Even though the percentages of different majors in the online and face-to-face samples are not all identical with each other, it does not mean either sample has an unfair advantage. First, this course has well-defined prerequisites that were enforced in all sections by the instructor. Second, the regression analysis in [8] showed cumulative GPA, not a student's major, was the single most important variable when the term grade was the dependent variable.

Table 3: Demographics: Percentages of Student Majors in the Online/Face-to-Face Sample

Student majors	Online Sample	Face-to-Face Sample
Computer Science (CS)	87%	75%
Computer Engineering (CpE)	10%	2%
Mathematics (MTH)	0%	14%
Others	3%	9%

Table 4 describes the background demographics of online and face-to-face (F2F) samples through the means of students' cumulative GPAs. Clearly, the cumulative GPA mean of the online sample is very close to that of the face-to-face sample, where the difference is only 0.62%. With α being 0.05, a Student's t-test was performed between all students' cumulative GPAs in the online sample and all students' cumulative GPAs in the face-to-face sample. The t-test results are also shown in Table 4. We can see that the p-value is much higher than 0.05 and the absolute value of the t-value is well below the two-tail T critical value. This indicates that there is no statistically significant difference between the cumulative GPA mean of the online sample and that of the face-to-face sample. We can conclude that neither the online sample nor the face-to-face sample has an unfair advantage in terms of student cumulative GPAs.

Table 4: Demographics: Cumulative GPAs (α = 0.05)

Online Mean	F2F Mean	*t*-Value	*T* Critical Value (two-tail)	*p*-Value	Significant?
3.22	3.20	0.1562	1.993	0.8763	N

The homework, exam, and term scores earned by every student in the online or face-to-face sample are used for student performance comparison. A student's *homework* or *exam score* is the percentage of the total homework or exam points this student earned in a semester. Using the same weights (50% from homework and 50% from exams) for calculating a student's final term grade, a student's *earned term score* is equal to 50% of this student's homework score plus 50% of this student's exam score. Any extra bonus point that was given to students for grade promotion, for some extra optional work, or for taking the survey is not included in any of a student's three scores for performance comparison.

Table 5 presents the results of two Student's t-tests (α = 0.05), one for comparing the homework scores of the students in the online sample with that of the students in the face-to-face sample and the other for comparing the exam scores of the students in the online sample with that of the students in the face-to-face sample. We can see that even though the homework score mean of the online sample is lower than that of the face-to-face sample and the exam score mean of the online sample is higher than that of the face-to-face sample, the p-values are higher or much higher than 0.05 and the absolute values of the t-values are both well below the two-tail T critical value. This indicates that there is no statistically significant difference between the performance of the students in the online sample and that of the students in the face-to-face sample in terms of homework and exam scores.

Table 5: T-Tests for comparing Homework/Exam Scores: Online vs. F2F (α = 0.05)

	Online Mean	F2F Mean	*t*-Value	*T* Critical Value (two-tail)	*p*-Value	Significant?
Homework	74.0%	79.1%	-0.9424	1.993	0.3491	N
Exams	68.9%	66.8%	0.3899	1.993	0.6977	N

To visually compare student scores in the online vs. face-to-face samples, Figure 3 gives the box-and-whisker plots of students' homework scores, exam scores, and earned term scores in the online sample and that in the face-to-face sample. We can see that, after excluding the outliers from each sample, the upper-quartile, median, and lower-quartile of the online sample are all very close to or higher than that of the face-to-face sample, respectively. This further shows that the student performance in the online section is statistically close to and not worse than the student performance in the face-to-face sections.

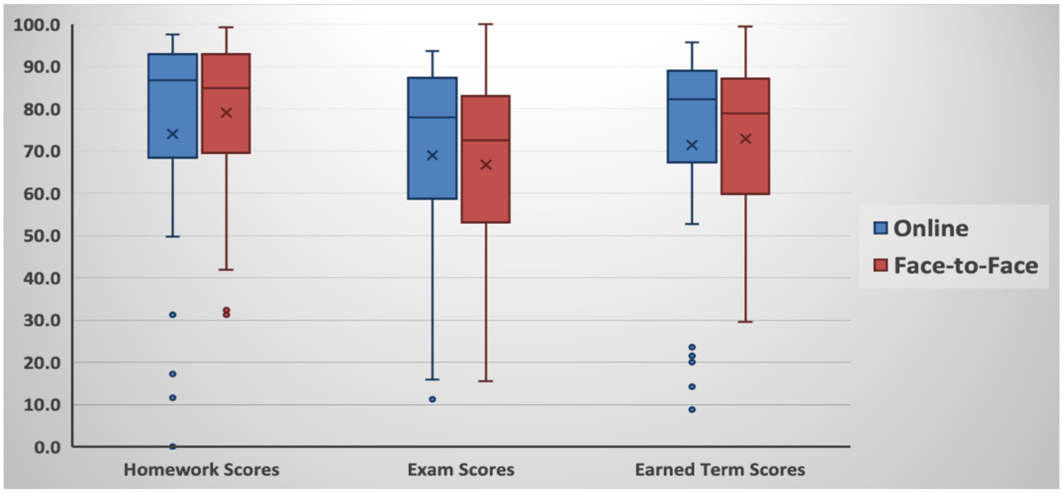

Figure 3: Box-and-Whisker Plots of Homework, Exam, and Earned Term Scores

In Figure 3, we can also see that the online sample has one or a few more outliers than the face-to-face sample. While there is no data available for further analysis because students took the survey anonymously and we don't have any info on students' personal situations, we would like to share our thoughts on possible causes. As discussed later in Section 5.2 and shown in Figure 4, unlike face-to-face students in Fall 2018 and Spring 2018, some (not all though) online students in Spring 2021 indeed were negatively impacted by the pandemic or related situations and/or faced technical difficulty to take lessons online at class time, where the negative impact or technical difficulty ranges between "low" and "high". Because lecture videos were recorded and posted, we think experiencing different levels of negative impact caused by the pandemic or related situations to students' time commitment to this course in Spring 2021 is a possible reason why the online sample has a few more outliers. Meanwhile, though not being officially reported in the survey, we did hear stories from both sides: a few students told us that not having a routine of going to classes daily made it harder for them to manage their time for learning activities while a few other students told us that the flexibility enabled by online teaching and learning made their learning more effective and efficient. Hence, another possible reason for the online sample to have a few more outliers is that the online teaching mode probably caused more diverse learning experiences and outcomes among students.

Nevertheless, we would like to point out that all student scores including those outliers' are used in calculating the mean values, *t*-values, and *p*-values presented in Table 5. Hence, having a few extra outliers in the online sample does not change our earlier conclusion that the performances of online students are not statistically significantly different from that of face-to-face students.

5.2 Analysis and results of a student survey in this online course

An anonymous student survey was designed to get students' perceptions of the online learning experience and their opinions on the instructional materials and learning activities in this online course. The first three questions (Q1 through Q3) are about the negative impact caused by the pandemic on students' availability during the class time or on their time commitment, or about the technical difficulty for taking lessons online during the class time. Followed are 15 Likert scale questions on the use of µVision 5, on instructional materials, and on hands-on learning activities. Table 6 shows a concise version of Q4 through Q18. Note that even though the statements of these 15 Likert-scale questions are positive, students were not misguided in this survey and were given the opportunity to truly express their opinions. First, students took the survey anonymously and Canvas does not reveal which answer was submitted by which student. Second, students were given the choices ranging from "strongly disagree" to "strongly agree".

Table 6: 15 Likert Scale Questions in the anonymous survey

	Questions in a concise version
Q4	Pausing/replaying lecture videos allows self-paced learning and is beneficial to learning.
Q5	Pausing/replaying videos allows following the instructor's demos to repeat hands-on activities.
Q6	Having access to lecture videos outside class time is helpful for *working on homework assignments*.
Q7	Having access to lecture videos is helpful *when not being able to attend class*.
Q8	The use of µVision 5 is helpful for improving the programming skills.
Q9	The use of µVision 5 allows debugging code in the same environment that the instructor uses for testing and grading.
Q10	Overall, *lecture videos* are useful for learning.
Q11	Overall, *lecture notes* are useful for learning.
Q12	Overall, *PPT slides* are useful for learning.
Q13	Overall, *example ARM programs* are useful for learning.
Q14	*Hands-on activities and labs* are helpful for achieving learning objectives.
Q15	*Programming assignments* are helpful for achieving learning objectives.
Q16	If it were F2F, you would still want *lecture videos* to be provided.
Q17	If it were F2F, you would still want *lecture notes* to be provided.
Q18	If it were F2F, you would still want *PPT slides and example programs* to be provided.

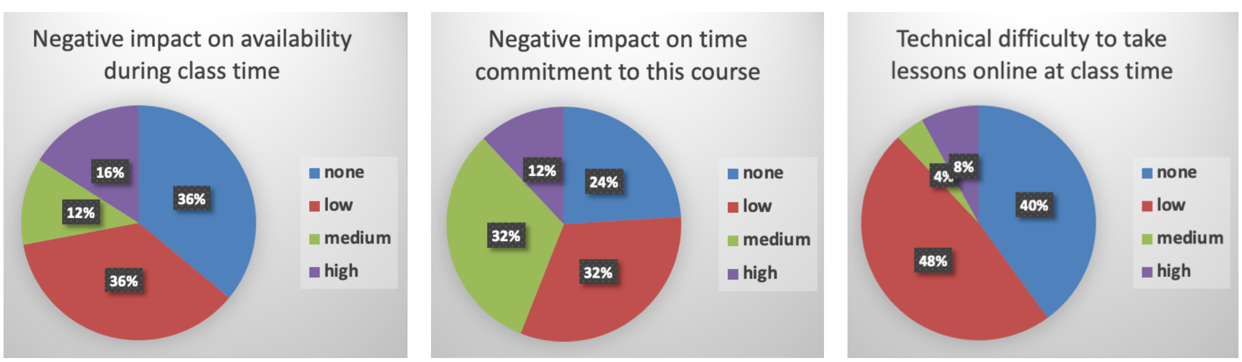

Figure 4: Ratings to Negative Impacts & Technical Difficulty

The anonymous survey in Canvas was given to students in this online course near the end of the Spring 2021 semester. We received the answers from 25 students. Figure 4 shows the percentages of students' ratings to the negative impact caused by the pandemic or its related situations on their availability during the class time (Q1), the negative impact caused by pandemic or its related situations on their time commitment to this course (Q2), and the technical difficulty to take lessons online at class time (Q3). It is shown that most of the students (ranging from 60% to 76%) were somehow negatively impacted by the pandemic or its related situations or faced some technical difficulty in this online course. Among these challenges, by average, the greatest one facing students is the negative impact caused by the pandemic or its related situations on their time commitment to this course.

Figure 5 plots the Likert scale ratings given by students to each of questions Q4 through Q18. While mapping every student's rating into a numeric value (5: strongly agree, 4: agree, 3: neither agree nor disagree, 2: disagree, 1: strongly disagree), the mean of the student ratings to every question is 4.3 or above. It is shown in Figure 5 that 100% of the students strongly agree or agree with Q6, Q10, Q11, and Q18; 92% to 96% of the students strongly agree or agree with Q4, Q5, Q7, Q9, Q12, Q15, Q16, and Q17; and 84% to 88% of the students strongly agree or agree with Q8, Q13, and Q14 while 68% of the students strongly agree with Q13. These results demonstrate a generally positive or very positive feedback from students regarding the use of µVision 5, regarding instructional materials, and regarding hands-on learning activities in this online course.

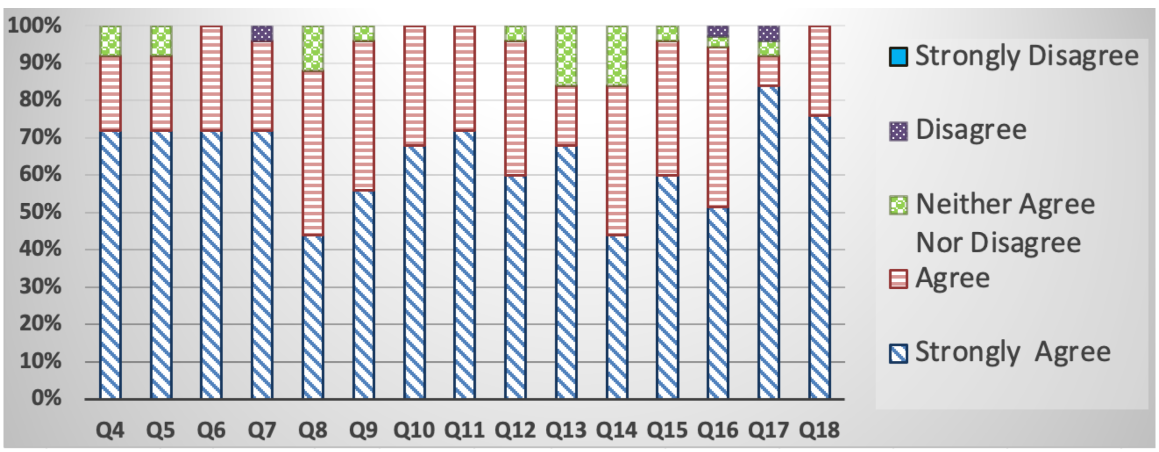

Figure 5: Likert-Scale Ratings to Q4 through Q18 in the Anonymous Student Survey

6 Conclusions and Future Work

We converted a traditional face-to-face assembly language and computer organization course into an online course. We adopted several technologies and developed various strategies to assure the smooth delivery of this online course and to support learner interactions and student learning assessment in this online course. Either the µVision IDE virtual app hosted inside our institutional private cloud together with the GlobalProtect client installed on a student's Mac computer or the µVision IDE installed on a student's Window computer was used for hands-on ARM assembly language programming. To evaluate the effectiveness of online teaching and learning, t-tests and box-and-whisker plots were used to compare online student performance with face-to-face student performance, and an anonymous student survey was given in the online course to get student perceptions and opinions.

We also found that the instructor did spend more time for class preparation teaching this course online than teaching it face-to-face. First of all, to accommodate the needs of some students whose availability were negatively impacted by the pandemic or related situations and some students who faced technical difficulty to take online classes during class time, the instructor spent a lot of time outside the class to pre-record lecture videos and post them in MS Stream. Secondly, to implement the strategies we developed to encourage academic integrity during exams, the instructor spent much more time making every online test than that would have been spent making a paper-based test. Simple questions on facts or basic concepts cannot be used any more. Multiple versions, instead of a single version, of every problem must be created and added to the question bank for this problem. Extra time were also spent comparing student work to check if there was any possible violation of academic integrity in an online test.

It is a little surprising but encouraging to find no loss in student performance in this online assembly language and computer organization course while comparing it to its face-to-face version. Given that there is no significant difference between online students' cumulative GPAs and face-to-face students' cumulative GPAs, t-test results show that there is not any statistically difference between the performance of online students and the performance of face-to-face students in terms of homework scores and exam scores. Box-and-whisker plots also indicate that the key statistics values of online student scores are close to that of face-to-face student scores. The greatest challenge, among three challenges asked in the survey, facing students in this online course is the negative impact caused by the pandemic or its related situations on their time commitment. Meanwhile, students' ratings to every Likert-scale question on their learning experience or the teaching in this online course are generally positive or very positive.

In the future, we will collect more assessment and survey data to replicate this research on a larger scale in this course or in other undergraduate computer science and computer engineering courses before we can further generalize our findings. Additionally, utilizing the lecture videos that we spent a lot of time making and specifically customized for the learning objectives and course contents of this assembly language and computer organization course, we will explore the flipped-classroom approach in this course, study its impact to student learning, and survey student perceptions and opinions. Moreover, it's interesting to see that online students in Spring 2021 did not statistically perform worse than face-to-face students in Fall 2018 and Spring 2018 in spite that some students' time commitment to this course was negatively impacted by the pandemic or related situations. To explore and explain why this happened, we plan to design and develop some new research method and use it to collect data in future semesters because our current method focuses on performance comparison instead of what causes the same or different performances.

REFERENCES

[1] I. Naimi-Akbar, L. Barman, and M. Weurlander, "Engineering teachers' approaches to teaching and learning online," In *Proceedings of the 2020 IEEE Frontiers in Education Conference (FIE), Uppsala, Sweden, October 21-24, 2020*. Piscataway, NJ: IEEE, 2020.

[2] P. Raviolo, "Online Higher Education Teaching Practices," In *Proceedings of the 10th International Conference on E-Education, E-Business, E-Management and E-Learning (IC4E '19), Tokyo, Japan, January 10-13, 2019*. New York, NY: ACM, 2019. pp. 79-84.

[3] D. C. Erdil, K. N. Bowlyn, and J. Randall, "A Hands-on Learning Approach to Introducing Computer Organization and Architecture to Early-college Students," in *2021 ASEE Annual Virtual Conference, July 26-29, 2021*. American Society for Engineering Education, 2021.

[4] C. Luo, "An Integrated Learning Approach Used in Computer Architecture," in *2019 ASEE Annual Conference & Exposition, Tampa, FL, June 15-19, 2019*. American Society for Engineering Education, 2019.

[5] B. Peterson and A. Clark, "PRISM: A Simple Simulation For Introduction Of Assembly Language And Computer Architecture," in *Proceedings of 2010 ASEE Annual Conference & Exposition, Louisville, KY, June 20-23, 2010*. American Society for Engineering Education, 2010.

[6] M. W. Redekopp and G. Ragusa, "Evaluating Flipped Classroom Strategies and Tools for Computer Engineering," in *Proceedings of 2013 ASEE Annual Conference & Exposition, Atlanta, GA, June 23-26, 2013*. American Society for Engineering Education, 2013.

[7] F. Yildiz, R. R. Pecen, and K. L. Coogler, "An Industry Related Project-Based Microcontroller Course," in *Proceedings of 2013 ASEE Annual Conference & Exposition, Atlanta, GA, June 23-26, 2013*. American Society for Engineering Education, 2013.

[8] M. McDonald, B. Dorn, and G. McDonald, "A Statistical Analysis of Student Performance in Online Computer Science Courses," *ACM SIGCSE Bulletin, vol. 36, no. 1*. New York, NY: ACM, March 2004. pp. 71–74.

[9] M. Whitney and B. Dallas, "Captioning Online Course Videos: An Investigation into Knowledge Retention and Student Perception," In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education (SIGCSE '19), Minneapolis, MN, February 27 - March 2, 2019*. New York, NY: ACM, 2019. pp. 511–517.

[10] S. Motogna, A. Marcus, and A.-J. Molnar, "Adapting to online teaching in software engineering courses," In *Proceedings of the 2nd ACM SIGSOFT International Workshop on Education through Advanced Software Engineering and Artificial Intelligence, Online, November 2020*. New York, NY: ACM, 2020.

[11] B. Benson and B. Mealy, "Experience with teaching digital design online," In *Proceedings of the 2014 IEEE Frontiers in Education Conference (FIE), Madrid, Spain, October 22-25, 2014*. Piscataway, NJ: IEEE, 2014.

[12] C. Wang and M. Goryll, "Design and Implementation of an Online Digital Design Course," in *Proceedings of 2016 ASEE Annual Conference & Exposition, New Orleans, LA, June 26-28, 2016*. American Society for Engineering Education, 2016.

[13] *Arm Compiler armasm User Guide*. Arm Limited or its affiliates, 2020. [E-book] Available: https://developer.arm.com/docs/dui0801/latest/

[14] D. G. Kleinbaum, L. L. Kupper, A. Nizam, and E. S. Rosenberg, *Applied Regression Analysis and Other Multivariable Methods (5th. ed.)*. Boston, MA: Cengage Learning, 2013.

[15] J. W. Turkey, *Exploratory Data Analysis (1st. ed.)*. Boston, MA: Addison-Wesley, 1977.