2026 ASEE ANNUAL CONFERENCE June 21–24, 2026 | Charlotte Convention Center, NC

Host Sponsor UNC Charlotte

Paper ID #51990

Foundations for a Minor in Quantum Computing at a Primarily Undergraduate Institution

Prof. David H. K. Hoe, Loyola University Maryland

David H. K. Hoe did his undergraduate and graduate studies at the University of Toronto. His professional experience includes positions at the General Electric Research and Development Center in Schenectady, NY as a Staff Engineer and at the University of Texas at Tyler as an Assistant Professor. He is currently an Associate Professor of Engineering at Loyola University Maryland. His research interests include high-performance computing, quantum computing, and engineering education.

Mary Lowe, Loyola University Maryland

Foundations for a Minor in Quantum Computing at a Primarily Undergraduate Institution

Abstract

Due to the rapid expansion of the quantum computing industry in recent years, there is an urgent need for trained workers in the field of quantum information, science, and technology. Primarily undergraduate institutions are a viable source of this much needed workforce, but they face unique challenges in developing comprehensive programs in quantum computing. To address this need, we have created a pair of courses in quantum computing suitable for students majoring in engineering, physics, chemistry, and computer science at primarily undergraduate institutions. These courses form the foundation for a minor in quantum computing that we are in the process of implementing at Loyola University Maryland. This paper describes our two interdisciplinary courses in quantum computing and our plans for implementing a minor in quantum computing.

1. Introduction

Quantum computing (QC) is one of the emerging technologies widely considered to have a revolutionary impact on the computing industry. In 1994, Peter Shor discovered that a quantum computer could theoretically factor large integers almost exponentially faster than the best classical algorithms [1]. Shor's algorithm, as it is now called, created the initial surge of interest in QC because of its potential to make current public-key encryption schemes obsolete. Since then, it's been shown that quantum computers have the potential for solving problems that are intractable on classical machines, such as the traveling salesman problem [2], make dramatic improvements in machine learning [3], and solve challenging problems in computational biology [4] and quantum chemistry [5-7]. Increased investments in quantum computing by both the government and private sector have brought about rapid developments in quantum hardware. As such, there is a critical need for engineers and scientists who have been trained in the field of quantum information, science, and technology (QIST). Not only is there a need for experts to design and build quantum computers and the associated quantum algorithms to run on these devices, but there is also a significant need for 'quantum-aware' engineers who will develop the supporting classical hardware and software that composes most of the quantum stack [8].

This work is primarily a curriculum and course design paper in which we describe the development and implementation of two undergraduate courses that will form the foundation for a minor in QC at Loyola University Maryland. Detailed assessments of our courses are being developed and will be reported in subsequent works. In this paper, we seek to provide insights on how primarily undergraduate institutions like ours can design a minor that is accessible to a wide range of STEM students. This paper will discuss the development of these two QC courses and highlight some of their unique features. For example, while our courses are lecture-based, a variety of active learning activities are integrated into each lecture. This includes activities where engineering, physics, and computer science (CS) majors work collaboratively in teams on in-class exercises, coding assignments, and projects. Such activities that emphasize student engagement during class-time are known to be more effective than traditional lectures [9, 10].

The interdisciplinary nature of these teams reflects the work environment in the QIST industry [11].

This paper is organized as follows. The second section gives some background on our institution and the impetus for developing these QC courses. The third section describes the implementation of our two QC courses, and the fourth section provides an overview of how our QC minor will be implemented for our engineering students. A concluding section provides a summary with strategies for institutions interested in implementing a minor in QC.

2. Background and Overview

Our school, Loyola University Maryland, is a private liberal arts institution that focuses primarily on undergraduate studies, with STEM departments in engineering, computer science, physics, chemistry, biology, and mathematics. As a primarily undergraduate institution (PUI), we understand the challenges of implementing courses in QC. This can include such issues as limited resources and lack of faculty expertise in QC. A recent survey of Quantum Information Science Engineering (QISE) education in the U.S. found that the majority of QC courses are concentrated in PhD-granting institutions while most institutions not offering a doctoral degree did not even offer a single such course [12]. In addition, most of the quantum and QISE courses tend to be offered at the graduate level. Hence, to meet the demand for more ‘quantum aware’ engineers, there is a real need to develop courses in QC that are suitable for undergraduate students.

The development of courses in QC at our institution occurred over several years. An initial attempt at teaching QC was carried out by a colleague in the physics department. He integrated some QC content into a Modern Physics course following a paper by Candela [13] and used the Mathematica package for implementing QC simulations. The second author subsequently did an independent study with three students on QC. The motivation and inspiration for teaching an introductory QC course had its genesis from a research project in the summer of 2021, in which we mentored two undergraduate students to develop a quantum computer simulator. The students created the simulator from scratch using the Python programming language and just the NumPy package. The success of this project led us to consider making coding an important part of our QC courses [14]. In the Spring 2023 semester, our introductory QC course was taught by the authors, representing the departments of engineering and physics, and another faculty member from the computer science department. We taught the introductory QC course again in the Fall 2024 semester. We then decided to add a second QC course in the Spring 2025 semester to give the students exposure to more advanced topics that focus on applications of QC.

3. Quantum Computing Courses

This section describes the two courses in QC that form the foundation for our proposed minor. The first course, Introduction to QC, provides the students with the basic background in QC circuits, while the second course covered specific applications, focusing on the quantum Fourier transform, Shor’s algorithm, and quantum machine learning. Both courses are designed for students majoring in engineering, physics, and computer science. With pre-requisites that can be taken in the freshman year, students at the sophomore level and above are able to enroll in the first course, which is the pre-requisite for the second course. This enables students to start taking the QC courses early in their college careers and makes our minor accessible to students that come from diverse educational pathways. Furthermore, by taking a cutting-edge course as soon as their second year, the expectation is that many students would be motivated to continue on in the STEM field.

3.1 First Quantum Computing Courses

The learning objectives for this course are as follows: at the completion of the first QC course we expect the students to be able to:

- describe basic concepts in quantum mechanics such as wavefunction (state), superposition, entanglement, and measurement
- perform the fundamental mathematical operations associated with quantum gates and qubits
- write computer code to implement the matrix operations associated with quantum gates
- articulate the advantages of quantum computing over traditional computing
- design, calculate the behavior of, and simulate simple quantum circuits
- describe the limitations of near-term quantum computing systems and effects of noise
- understand basic quantum computing algorithms such as quantum adders, Deutsch's algorithm, the Deutsch-Jozsa algorithm, and Grover's algorithm.

The topics in our introductory QC course were chosen to ensure the students are able to understand the operation of basic quantum gates, can analyze simple quantum circuits, and have an understanding of key quantum algorithms.

The mathematics involved in our Introduction to QC course is relatively simple, requiring only a good understanding of high school algebra and trigonometry. The prerequisites are an introductory programming course and Calculus I. The latter ensures the student is able to handle the pace and learning expectations of a college level STEM course. No background in introductory college physics or quantum mechanics is required as most of the physics topics required for understanding QC are not covered in those courses. The required mathematics background, such as complex numbers, probability, and linear algebra, is introduced as needed. Only a subset of what is typically taught in a linear algebra course is required to understand QC at an introductory level. For example, see textbooks [15, 16]. Some of the mathematical notation used in QC are a challenge for the students to learn. However, we introduce some of the essentials such as Dirac's bra-ket notation as it is a key to analyzing and understanding quantum circuits, and we want our students to eventually have the ability to read the scholarly literature on QC. When introducing the various quantum algorithms, we strived for a conceptual understanding and selected aspects of the mathematical formalism and proofs. More advanced algorithms such as the quantum Fourier transform and Shor's algorithm are deferred to our follow-up QC course.

Traditional lectures were used where students were provided with scaffolded notes. Space is provided for the students to write notes and embedded in the notes are short exercises that were worked on in class. When completing the exercises, students were often encouraged to discuss the solution with their neighbors. Weekly homework assignments consisted of both paper-and-pencil exercises as well as writing Python/NumPy or Qiskit code. This combination of traditional exercises and code writing ensures the students were assimilating the material. A range of active-learning activities were incorporated into our lectures, from simple in-class exercises and think-pair-share activities that lasted a few minutes to cooperative learning activities that could take the whole class time. Several topics were introduced using the cooperative learning method known as Jigsaw [17], which has recently been adapted for use in engineering courses [18–21]. Students worked in teams of four on the Jigsaw exercises, which are structured so that they must work collaboratively to successfully complete the exercises. Table I summarizes the Jigsaw exercises that have been implemented.

Table I. Summary of Jigsaw Activities

Week	Topic
2	Visualizing the Bloch Sphere
5	Coding Exercises: Measuring a Quantum State
7	Deutsch's Algorithm
10	Superdense Coding
11	Quantum Adders
12	Constructing Gate Matrices via Outer Products

In the first iteration of the course, we tasked the students with building their own Python-based simulator from scratch using object-oriented programming. Students worked in pairs for the first three assignments and then in groups of four for the final two assignments, where each team had at least one CS student in it. While most of the groups were able to complete the simulator, we found that the CS students were doing most of the work. Our original thinking was that the engineering and physics students would be helpful in explaining the mathematical theory, but at least in this initial class, the CS students were mostly able to understand the mathematics as well. Hence, for the second iteration, we simplified the Python coding assignments and had the students primarily work with Qiskit to simulate more complex quantum gates. We found with this approach the students were still able to meet the learning requirements of the course and there was a lot less stress and complaints from the students.

In the last three weeks of the semester, students worked on a course project representing some open problem in quantum computing. The students could work individually or in pairs. There were two requirements: first, doing a review of the relevant published literature on the topic, and second, performing some simulations using the quantum computer simulator developed in the course. The course project provided the students with authentic research experience, albeit it was an abbreviated one, in QC. The project also gave the students an opportunity to investigate an area of interest beyond what was discussed during the lectures. The projects covered a range of topics of interest in QC, from using parallel computation and sparse matrix memory methods to improve the simulation of quantum computers to studying the issue of noise in quantum computers and investigating novel adder structures that can run on QCs. We have discussed ways for undergraduate students to have a significant role in authentic research experience in QC projects in [23]. Overall, the students benefited from working on problems that go beyond standard textbook questions and provided insights, albeit preliminary given the time constraints, to the important issues in QC.

3.2 Second Quantum Computing Course

Upon surveying our students from the first QC course, we observed that the students generally did not have a good sense of the advantages of QC and its potential applications. Hence, we decided it would be beneficial to the students to add a second course, which was taught for the first time in the spring 2025 semester. The focus was on the quantum Fourier transform (QFT), Shor's algorithm, and quantum machine learning. The only prerequisite for this second course was the Introductory QC course. Any of the required background, such as the Fourier transform and an introduction to classical machine learning, is provided in the QC II course. Students from engineering, physics, and computer science who were interested in the introductory course are likewise the targets of the second course. The learning objectives for this course are as follows: at the completion of the course, we expect the students to be able to

- describe the quantum Fourier transform (QFT) and its application in quantum algorithms
- understand how Shor's algorithm can be used to factor numbers
- describe how quantum phase estimation is used in quantum algorithms
- understand how data is encoded as quantum states
- articulate some approaches for implementing machine learning on quantum computers
- write computer code to simulate various quantum machine learning algorithms
- understand the issues that must be overcome for QML algorithms to be implemented at a practical scale on QC platforms
- describe some approaches for quantum error correction
- be able to read some of the scholarly literature in quantum computing for doing research

The QFT algorithm is foundational for QC and is used in Shor's algorithm. After studying Shor's algorithm, the students simulated a simplified 5-qubit version of the algorithm designed to factor the number 15 [24]. They also studied two other algorithms that make use of quantum phase estimation (QPE), the inverse of the QFT. The first was a proposal to solve an optimization problem known as the Travelling Salesman Problem [2]. This involved analyzing an implementation for solving the problem with four cities. The second algorithm using QPE was the Harrow–Hassidim–Lloyd (HHL) algorithm [25], which enables a system of linear equations to be solved and is central to many machine learning methods. To understand the HHL algorithm, students learned about Hamiltonian simulation and the Spectral Theorem for matrices. Students coded a simplified version of the algorithm and simulated it in Qiskit with the help of a set of tutorials that outlined the required code.

Quantum machine learning (QML) is a challenging subject for students to understand because it requires a knowledge of classical machine learning methods as well as a solid understanding of QC concepts. For the machine learning (ML) part of the course, we needed to introduce the students to basic concepts in classical ML first. After describing unsupervised and supervised ML approaches, an overview of kernel-based approaches was undertaken. The majority of ML approaches in general can be considered to make use of the kernel concept and there are some unique features of QML that make a kernel-based approach attractive [26]. After describing the kernel-based approach in a classical sense, we studied a QML approach that implements a very simple distance-based classifier [27]. Although only a simplified dataset could be tested, the circuit demonstrates the advantage of quantum parallelism while also showing the bottleneck created when classical data must be encoded onto a quantum machine. Finally, brief overviews of quantum neural networks (QNNs) and quantum error correction were given [28].

Similar to the first course, the students were provided with scaffolded notes and lectures were primarily used. Python coding tutorials were given to introduce key concepts in machine learning and students were given some time in class to work on them and to interact with the instructor. As most of the faculty time was spent developing the content for the lectures, this first iteration of the course did not use any team-based learning activities. Like the first QC course, the students worked on a course project during the last three weeks of the semester. The projects covered some area of research in QML, which included such topics as quantum Bayesian ML, quantum recurrent neural networks, quantum generative adversarial networks, and quantum autoencoders. Students were required to do a literature search and find several recent papers on their topic and to report on the state-of-the-art proposals or theoretical developments. They were also asked to do some simulation on the QML algorithm in order to gain a better understanding of how it works. Students did a short presentation to the class summarizing their findings and turned in a written report. Overall, it appeared the students benefited from investigating a specific area of research in QML and reporting it to their classmates as it provided added motivation to learn the material.

3.3 ABET Assessment of the Courses

In this subsection, we provide assessments indicating the course learning objectives are being met based on our course evaluations which were prepared for ABET. This includes student self-assessment data and faculty evaluations. More detailed assessments are being developed and will appear in future works [22]. The student self-assessment data was obtained from Qualtrics surveys administered at the end of the semester. There were 14 and 16 students responding from the 2023 and 2024 Intro to QC courses, representing 70% and 84% of the students in the classes, respectively. Students were asked to rate their agreement with meeting each learning outcome on a 5-point (Likert) scale where 5 = strongly agree, 3 = neutral, and 1 = strongly disagree. For the faculty assessment, student performance as reflected in student work was evaluated by the instructors using the following 5-point scale described in Table II. Student work spanning the range of course assignments and academic performance is examined. Special attention is paid to C-level work as this represents minimum competence.

Table II. Faculty Course Assessment Criteria

	Much Lower than Expected 1	**Somewhat Lower than Expected 2**	**As Expected 3**	**Somewhat Higher than Expected 4**	**Much Higher Than Expected 5**
Description	No students showed an acceptable level of understanding	A majority of students did not show an acceptable level of understanding	Top students showed mastery, and C-level students showed an acceptable level of understanding	A majority of students showed mastery, and essentially all remaining students showed an acceptable level of understanding	All students showed mastery

The following table summarizes the results of the Spring 2023 and Fall 2024 offerings of the Introductory QC course.

Table III. Assessment of the Introductory QC Course

Learning Outcome	**Student**			**Faculty**	
	2023	**2024**		**2023**	**2024**
1. describe the basic concepts in quantum mechanics	4.36	4.38		3	3
2. perform the fundamental mathematical operations associated with quantum gates and qubits	4.64	4.69		3	3
3. write computer code to implement the matrix operations associated with quantum gates	4.29	4.50		3	3
4. articulate the advantages of quantum computing over traditional computing	4.07	4.44		3	3
5. write code to make measurements, display them, and interpret the results	4.13	3.81		3	3
6. design and simulate simple quantum circuits	4.57	4.31		2	3
7. describe the limitations of near-term quantum computing systems and effects of noise	3.64	3.75		1	2
8. list some important areas where quantum computing can make a contribution	4.00	4.25		3	2
9. describe qualitatively several examples of quantum computing hardware and their physical principles	3.47	3.69		1	1

The assessments indicate that most of the learning outcomes were met. Learning outcomes 7 and 9 did not meet the faculty expectations mainly because there was insufficient time to fully cover these topics. The other room for improvement was learning outcome 6 in 2023. We addressed this issue by simplifying the coding assignments with improved guidance in the tutorials in subsequent course iterations. One other issue we observed is that students generally found working with the notation of QC (e.g., Dirac's bra-ket formalism) to be challenging and preferred doing analysis via matrices instead. We are addressing this by placing more emphasis on working with the notation in the current iteration of the first course.

Table IV summarizes our ABET assessment for the second QC course. Six students, representing 67% of the class, responded to the end of the semester Qualtrics survey. As this was the initial offering of the course, there are several things that can be improved. Two main observations are given here. First, more time was spent on the front end of the course than was initially planned, especially the quantum Fourier transform (QFT). This left little time to cover the material at the end, especially quantum error correction (hence, the rating of 1 for it). Second, the students found it challenging to be able to read and comprehend the scholarly literature. This is a consequence of the advanced nature of the course and can be remedied in part by covering more of the backend material by reducing the time spent on the QFT.

Table IV. Assessment of QC II Course

Learning Outcome	Student		Faculty
	2025		2025
1. describe the quantum Fourier transform and its application in quantum algorithms	4.33		3
2. understand how Shor's algorithm can be used to factor numbers	4.67		3
3. describe how quantum phase estimation is used in quantum algorithms	4.43		3
4. understand how data is encoded as quantum states	4.83		4
5. articulate some approaches for implementing machine learning on quantum computers	4.43		3
6. write computer code to simulate various quantum machine learning algorithms	4.50		3
7. understand the issues that must be overcome for QML algorithms to be implemented at a practical scale on QC platforms	4.67		3
8. describe some approaches for quantum error correction	4.00		1
9. be able to read some of the scholarly literature in quantum computing for doing research	4.00		2

From the Spring 2023 cohort of students, we have some anecdotal evidence of the impact of the course. One of the students has gone on to pursue a doctorate at the University of New Hampshire in the area of post-quantum cryptography, effectively leveraging her background in QC. In addition, a couple students reported that they mentioned having taken the Intro to QC course in job interviews, which has made them stand out as candidates.

3.4 Future Plans to Improve the Quantum Computing Courses

We currently have a three-year NSF IUSE grant to develop active learning material for both QC courses. We are currently working on creating such material, which includes adding more medium-length (20 to 30 minute) team-based exercises and assessing and improving our five cooperative learning (Jigsaw) exercises for the first course this spring 2026 semester. In addition, we will add team-based active learning activities to our second course. A module on QC applied to quantum chemistry will be added to the second QC course as well. All of this material will be assessed by an external evaluator, and our course content and assessment findings will be disseminated to interested college instructors.

4. Our Proposal for a QC Minor

The purpose of the proposed QC minor is to create graduates capable of entering the workforce as 'quantum-aware' engineers and scientists – they will have sufficient background to design the classical circuits and computer software that support quantum computers. Six courses comprise the minor. There are four required courses: Introduction to QC and QC II are the foundation courses together with modern physics and linear algebra. The two remaining courses are electives drawn from senior level courses. In this paper, our focus is on the electrical and computer engineering curriculum, although we are also proposing elective courses that are geared towards students majoring in physics and CS.

For the objectives of our QC minor, we expect a student will be able to:

1. understand the key theoretical principles of QC (superposition, entanglement, interference, measurement, ancilla, oracle, gates, etc.).
2. be able to write code and use software packages to simulate quantum circuits
3. understand the important quantum algorithms: Deutsch-Jozsa, Grover, QFT, Shor's, and the HHL algorithm
4. be knowledgeable of key applications of QC in machine learning and QC applied to quantum chemistry

The Introduction to QC course mostly covers the first two objectives and part of the third (Deutsch-Jozsa and Grover's algorithm) and the QC II course covers the QFT, Shor's and HHL algorithms and the fourth objective. The Modern Physics course covers Einstein's special theory of relativity and quantum physics. The Linear Algebra course is important for reinforcing concepts in vector spaces, matrix operations, and eigenvalues. Table V lists some of the elective courses.

Table V. Elective Courses for the QC Minor

Course Name	Description
EG 475 FPGA Design	Programming of FPGAs, understanding their technology, and coding with VHDL
EG 477 VLSI Design	Design, layout, characterization, and testing of integrated circuits
EG 483 Control Systems	Analysis and design of feedback control systems
EG 485 Digital Signal Processing	Discrete-time systems, convolution, discrete-time Fourier transform, digital filter design, z-transform
EG 489 Special topic in electrical engineering	With a QC topic or machine learning topic
EG 495 Independent Research Study	On a QC topic

High-precision classical electronics for controlling and measuring individual qubits are a critical part of the quantum stack. Hence, quantum-aware engineers need the background provided by the engineering elective courses that make up this minor. The Field Programmable Gate Array (FPGA) Design course is important as FPGA technology is typically used for processing the data and implementing the control circuitry [29–31]. The Control Systems and Digital Signal Processing courses provide the theoretical background needed for designing the high-precision timing circuits and signal processing modules on the FPGAs. The VLSI design course provides the relevant background in integrated circuit (IC) design as the technology used to fabricate microchips is similar to the processing techniques used to build superconducting qubits. It is noteworthy that Intel is leveraging their IC technology that is used for creating their microprocessors in the building of silicon spin qubits. They recently announced a 12 qubit QC chip [32]. Students are also able to select several math courses as electives. For example, math courses in group theory and number theory can be used as electives as they are relevant for understanding encryption technologies – i.e., Shor's algorithm is able to break current RSA based encryption schemes and some background here provides insight into post-quantum encryption schemes. Other useful elective math courses include Advanced Linear Algebra and Numerical Simulation.

Our minor allows an overlap of up to four courses with the courses in an engineering major so that a student is required to take two more courses than typically is required to get an engineering degree. This is illustrated in the Venn diagram in Figure 1. Hence, students can realistically complete their engineering degree and the QC minor within a four-year period.

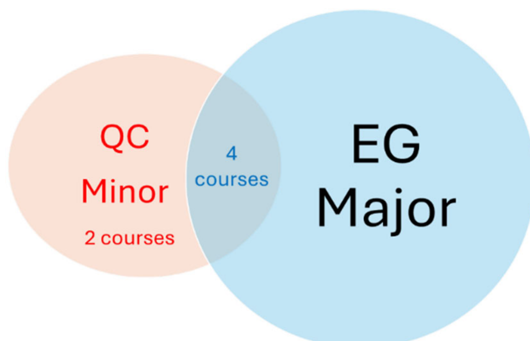

Figure 1. Venn diagram illustrating the overlap in courses between the engineering (EG) major and the QC minor.

Table VI summarizes the typical sequence of courses for an EE student enrolled in the minor. While a student can take our Introduction to QC course as early as the fall sophomore year, most typically take it in their junior year due to the way our engineering courses are structured.

Table VI. Typical Course Sequence for an EE Student

Course Name	Typical Semester Taken
PH 312 Modern Physics	Fall – Sophomore
EG 488 Introduction to QC	Fall – Junior
EG 492 QC II	Spring – Junior
MA 301 Intro to Linear Algebra	Spring – Junior
EG 475 FPGA Design	Fall – Senior
EG 477 VLSI Design	Spring – Senior

There are not many schools offering minors that focus specifically on QC for undergraduates. There are some programs that feature programs and minors in quantum engineering. Our survey of PUIs found only one other institution that is currently offering a minor in QC. In reviewing the existing minors at other institutions, we found our proposal to be in line with other offerings. Our proposal for two dedicated courses in QC, plus a range of supporting courses in electrical and computer engineering, CS, and physics, is typical of the requirements for a minor in QC found at other schools.

5. Summary and Conclusions

This paper has described two courses in QC and outlined a proposed minor in QC. From our experience, we offer the following strategies that other PUIs can use to develop a similar minor in QC.

1. To gain experience in the area of QC, consider initially adding a module or two into existing courses. This incremental approach is relatively straightforward to implement and allows one to gradually gain knowledge in the area. We also found it helpful to mentor undergraduate students on summer projects in QC. This gave us opportunities to develop tutorial materials and to test out its effectiveness on the students. There are several short courses available online and through Zoom that enable one to learn the basics of QC.
2. As one becomes comfortable introducing several modules into existing courses, adding an introductory course in QC becomes realistic. Consider making this course available across several disciplines. This ensures sufficient enrollments for smaller schools and allows several faculty members to support the course. A potential challenge is gaining the support of your college to be able to team teach the course. We were able to teach our initial offering of the introductory QC course with faculty members from engineering, physics, and CS.
3. To complete the curriculum for the minor, develop the second course which can include more advanced content along with specific applications in QC. To encourage enrollment in the second QC course, we have found it helpful to offer a certificate or microcredential in QC. Students taking our two QC courses plus an introductory programming course can earn a microcredential in QC at our school. Again, tying the course content to related research projects with undergraduate students over the summer can be helpful in developing teaching material as we were able to do with our module on QML.

For our three offerings of the introductory QC course, we have consistently been enrolling 18 to 20 students. For our QC II course offered in the Spring 2025 semester, there were 9 students enrolled. Based on these enrollments and anecdotal feedback from our students, we estimate that 4 to 6 students could initially be enrolled in our QC minor program with the numbers increasing as the program develops and its potential for preparing students for the quantum industry becomes more widely known.

The development of a QC minor at PUIs will help meet the increased demands for engineers and scientists trained in QIST due to the continuing growth of the QC industry in recent years. Our proposed QC minor is suitable for undergraduate students and aims to produce trained graduates able to function as ‘quantum aware’ engineers, physicists, and computer professionals. Our teaching material aims to improve student engagement and learning in the classroom, while lowering the on-ramp to enable students to access our QC courses from diverse educational backgrounds. With funding from an NSF grant, we will be assessing our curriculum and disseminating our approach to interested college educators from across the U.S.

References

1. P. W. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," *Proceedings of the IEEE 35th Annual Symposium on Foundations of Computer Science*, pp. 124–134, Nov. 1994.
2. K. Srinivasan, S. Satyajit, B. K. Behera, and P. K. Panigrahi, "Efficient quantum algorithm for solving travelling salesman problem: An IBM quantum experience," *arXiv preprint arXiv:1805.10928*, 2018.
3. Y. Wang and J. Liu, "A comprehensive review of quantum machine learning: from NISQ to fault tolerance," *Reports on Progress in Physics*, vol. 97, no. 11, p. 116402, 2024.
4. C. Outeiral, *et al*., "The prospects of quantum computing in computational molecular biology," *Wiley Interdisciplinary Reviews: Computational Molecular Science*, vol. 11, no. 1, p. e1481, 2021.
5. Y. Cao, *et al*., "Quantum chemistry in the age of quantum computing," *Chemical reviews*, vol. 119, no. 19, pp.10856-10915, 2019.
6. B. Bauer, *et al*., "Quantum algorithms for quantum chemistry and quantum materials science," *Chemical Reviews*, vol. 120, no. 22, pp. 12685–12717, 2020.
7. S. McArdle, *et al*., "Quantum computational chemistry," *Reviews of Modern Physics*, vol. 92, no. 1, p.015003, 2020.
8. A. Asfaw, *et al*., "Building a quantum engineering undergraduate program," *IEEE Transactions on Education,* vol. 65, no. 2, pp. 220–242, 2022.
9. M. Prince, "Does active learning work? A review of the research," *Journal of engineering education*, vol. 93, no. 3, pp. 223–231, 2004.
10. D. Lombardi, *et al*., "The curious construct of active learning," *Psychological Science in the Public Interest*, vol. 22, no. 1, pp. 8–43, 2021.
11. C. Hughes, *et al*. "Assessing the needs of the quantum industry," *IEEE Transactions on Education*, no. 65, no. 4, pp. 592–601, 2022.
12. A. R. Piña, *et al*, "Landscape of quantum information science and engineering education: from physics foundations to interdisciplinary frontiers," *Physical Review Physics Education Research*, vo1. 21, no. 2, p. 02013, 2025.
13. D. Candela, "Undergraduate computational physics projects on quantum computing," *American Journal of Physics*, vol. 83, no. 8, pp. 688–702, 2015.
14. D. H. Hoe, M. Lowe, D. Binkley, "Work-in-progress: Undergraduate courses in quantum computing: a proposal based on our experience building a Python-based quantum computer Simulator," in *2022 ASEE Annual Conference & Exposition*, 2022.
15. N. Yanofsky and M. Mannucci, *Quantum Computing for Computer Scientists*, Cambridge: Cambridge University Press, 2008.
16. T. L. Wong, *Introduction to Classical and Quantum Computing*, Rooted Grove, 2022.
17. E. Aronson, *The Jigsaw Classroom*, SAGE Publications, 1978.
18 J. K. Yousafzai, I. Damaj, and A. K. Yousafzai, "Incorporating jigsaw cooperative learning in a signals and systems course," in *2017 IEEE Global Engineering Education Conference,* pp. 1225–1228, April 2017.
19. J. A. Pow-Sang and P. Escobar-Cáceres, "A systematic literature review of the application of the jigsaw technique in engineering and computing," in *International Conference on Interactive Collaborative Learning*, pp. 322–329, 2017.

20. M. C. Marinescu and J. L. Marzo, “Using cooperative teaching techniques in engineering courses. The Jigsaw case,” *International Association for Development of the Information Society*, 2019.
21. M. Minetti, “A Jigsaw approach for electrical engineering: transient stability analysis of a synchronous machine,” *2024 IEEE International Conference on Environment and Electrical Engineering and 2024 IEEE Industrial and Commercial Power Systems Europe,* pp. 1–6, June 2024.
22. D. H. K. Hoe, M. Lowe, D. Binkley, “An introductory undergraduate course in quantum computing: an interdisciplinary, coding-based approach,” *manuscript in preparation*.
23. D. H. Hoe, M. Lowe, D. Binkley, “Undergraduate research in quantum computing: lessons learned from developing student researchers,” in *2023 ASEE Annual Conference & Exposition*, 2023.
24. J. Abhijith, *et al*. “Quantum algorithm implementations for beginners,” *ACM Transactions on Quantum Computing*, vol. 3, no. 4, pp. 1–92, 2022.
25. A. W. Harrow, A. Hassidim, and S. Lloyd, “Quantum algorithm for linear systems of equations,” *Physical review letters*, vol. 103, no. 15, p. 150502, 2009.
26. M. Schuld, “Supervised quantum machine learning models are kernel methods,” *arXiv preprint arXiv:2101.11020*, 2021.
27. M. Schuld, M. Fingerhuth, and F. Petruccione, “Implementing a distance-based classifier with a quantum interference circuit,” *Europhysics Letters*, vol. 119, no. 6, p.60002, 2017.
28. J. Roffe, “Quantum error correction: an introductory guide,” *Contemporary Physics*, vol. 60, no. 3, pp. 226–245, 2019.
29. I. D. Conway Lamb, *et al*. “An FPGA-based instrumentation platform for use at deep cryogenic temperatures,” *Review of Scientific Instruments*, vol. 87, no. 1, 2016.
30. Y. Xu, *et al*., “QubiC: An open-source FPGA-based control and measurement system for superconducting quantum information processors,” *IEEE Transactions on Quantum Engineering*, vol. 2, pp. 1–11, 2021.
31. N. Irtija, J. Plusquellic, E. E. Tsiropoulou, J. Goldberg, D. Lobser, and D. Stick, “Design and analysis of digital communication within an SoC-based control system for trapped-ion quantum computing,” *IEEE Transactions on Quantum Engineering*, vol. 4, pp. 1–24, 2023.
32. Intel Labs: Discover Quantum Computing, https://www.intel.com/content/www/us/en/research/quantum-computing.html [Accessed Jan. 20, 2026]