Course Syllabus – Fall 2025

CPSC 4470 - Introduction to Quantum Computing (cross-list CPSC 5470)

Website: https://www.yongshanding.com/cpsc447-f25/

Instructor: Prof. Yongshan Ding

Short Description: This course will introduce the fundamental concepts in the theory and practice of quantum computation. Topics covered include information processing, quantum programming, quantum compilation, quantum algorithms, and error correction. The objective of the course is to engage students in applying a fresh thinking to what computers can do – we establish an understanding of how quantum computers store and process data, and discover how they differ from conventional digital computers.

Prerequisites: After CPSC 2010 and CPSC 2020. Basic familiarity with discrete probability and linear algebra is recommended. Prior experience in quantum computing is useful but not required. We anticipate this course will be of interest to students working in computer science, electrical engineering, physics, or mathematics.

Recommended Textbooks and References:

1. Quantum Computer Systems, by Ding and Chong. Free access via Yale VPN [https://link.springer.com/book/10.1007/978-3-031-01765-0].
2. Introduction to Quantum Error Correction and Fault Tolerance, by Girvin [https://arxiv.org/abs/2111.08894]
3. Quantum Computation and Quantum Information, by Nielsen and Chuang.

Lectures: This course will be an introduction to quantum computing and information. Classes meet on **MW 11:35 AM-12:50 PM**, in **ML 211**. The lectures are divided into several modules:

- Module 1. Introduction to the principles of quantum computation:
    a. Quantum Computing – A CS Perspective
    b. Quantum States: Superposition for Computation
    c. Quantum Operations and Circuits
    d. Quantum Entanglement
    e. Quantum Teleportation
    f. No-cloning Theorem
- Module 2. Quantum Logic and Programming
    a. Program Compilation
    b. Reversible Logic
    c. Quantum Oracles
- Module 3. Quantum Algorithms
    a. Bernstein-Vazirani Algorithm, Deutsch-Jozsa Algorithm
    b. Grover's Search Algorithm, Amplitude Amplification
    c. Simon's Algorithm, Quantum Phase Estimation
    d. Variational Algorithms
- Module 4. Noisy Quantum Systems
    a. Quantum Channels, Operator Sum Representations

b. Quantum Fault Tolerance

Evaluation: The course grade will be based on online and in-class discussions (20% for CPSC 4470, 15% for CPSC 5470), three assignments (30%), two in-class midterm exams (30%), and a final exam (20%). Additionally, students in CPSC 5470 will conduct a project on selected topics (5%).

Class Participation: Learning happens best through interaction. In addition to in-class discussions, each class will be followed by an online discussion (on Ed) about the day's topics. Participation will be evaluated based on both the quality and quantity of contributions.

Assignments: Assignments are due on Gradescope at 11:59PM US Eastern Time unless otherwise indicated on the assignment. Each student has a total of **4 late days** to be used over the entire semester. No more than two late days can be used on a single assignment. If you do not have remaining late days, late submissions will incur a 15% penalty per day. No assignments will be accepted more than two days after their original deadline. The four assignments will have written and/or programming components on the following topics:

- Assignment 1: Basics of Quantum Computation. (10%)
- Assignment 2: Advanced Quantum Circuits. (10%)
- Assignment 3: Fun with Quantum Algorithms. (10%)

Exams: There are three exams in this course:

- Midterm Exam I in class on Monday, **October 6th**. (15%)
- Midterm Exam II in class on Monday, **November 3rd**. (15%)
- Final Exam in class on Wednesday, **December 3rd**. (20%)

Collaboration and Academic Integrity: To facilitate learning, collaboration on the assignments is encouraged. A discussion may take place in a small group (on a whiteboard or on a scratch paper, etc.), but notes or recordings of the discussion are prohibited. All assignments should be written up on one's own after sufficient time (i.e., at least two hours) has elapsed since any discussion. You must list all your collaborators on every assignment. If you have used external reference materials, please also note this on the assignment. The objective is to demonstrate that you can (re)create a solution on your own. It is prohibited, however, to share solutions or provide hints to other students after you have already arrived at the solution. Take a moment to familiarize yourself with Yale's Academic Integrity Policy (http://catalog.yale.edu/undergraduate-regulations/policies/definitions-plagiarism-cheating/) if you have not done so already. If you have any questions about what does or does not constitute plagiarism, ask! Plagiarism is a serious offense and will not be treated lightly. Fortunately, it is also easy to avoid and if you are the least bit careful about giving credit where credit is due you should not run into any problems.

Generative AI Tools Policy: Unless stated otherwise, you're welcome (and even encouraged) to use generative AI (GAI) for course coding and projects, as long as you give credit. You can use AI tools to help you generate code or designs, or to polish code you've already written. Just remember: these tools aren't perfect and can make mistakes or "hallucinate." That's why every submission, AI-assisted or not, needs to be tested extensively and you are responsible for what you submit. For non-programming assignments (like the written component), GAI tools are *not* allowed. Your response should reflect your own thinking and writing. When in doubt, don't hesitate to reach out!