Lesson Planning: Introduction to Quantum Computing

Talent and Learning Center
Amsterdam University of Applied Sciences
Email: qtlcamsterdam-ft@hva.nl

3rd draft: April 3, 2025

Introduction to Quantum Computing Workshop

This workshop is designed to introduce students to the basics of quantum computing and quantum software in a simple, interactive, and engaging manner with a focus on hands-on learning.

Workshop information

Teaching style

- Short theoretical explanation and teaching by doing.
- Hands-on exercises and games to reinforce learning.

Target audience

- First- and second-year undergraduate students of engineering, computer science, physics, and mathematics at applied science universities.

No background knowledge required

- No prior knowledge of quantum computing, quantum physics, linear algebra, or complex numbers is required.
- No prior programming experience is required.

Duration and planned schedule

- 6 lessons, each lasting 3 hours.
- Held in quarter 4 of the academic year starting in April and ending in June.
- 1 lesson per week, every Friday, starting 14:00 and ending 17:00.
- Days of the workshop (proposal): April 25th, May 9th, May 16th, May 23rd, June 6th, and June 13th.

What students will learn

- Why quantum computing is needed.
- Basic principles of quantum computing.
- Mathematical tools for quantum computing.

- Linear algebra and quantum computing.
- Introduction to Qiskit.
- Hands-on experience with quantum computing using Qiskit.
- How to run simple quantum systems using Qiskit.
- Quantum computing in encryption.
- Run quantum programs on real quantum computers like Delft's Quantum Inspire.

Your Quantum Journey Starts Here

- Students can apply for Applied Quantum Computing minor at Amsterdam University of Applied Sciences.
- Also, after the bachelor's degree, students can apply for Applied Quantum Technology master's degree in quantum software engineering.

The workshop will cover the following topics:

Lesson 1: Why Do We Need Quantum Computing?

Topics:

- Why quantum and what is quantum computing?
- How does it work?
- Introduction to applications.
- Classical vs qubit: superposition, entanglement, measurement.
- Experimental examples of quantum computing if available.

Games: Quantum tiq taq Toe (beginner level) and Psi and Delta

Learning Objectives (LOs):

- Understand why we need quantum computing.
- Understand the basic principles of quantum computing.

Lesson 2: Fun with Qubits

In this lesson, students will learn about qubits, the basic unit of quantum information, and the necessary mathematical tools to understand quantum qubits and be able to manipulate them in Qiskit, the quantum software development kit.

Topics:

- **What is a Qubit?**
  - Introduction and visualization of a single qubit.
- **Interactive Bloch Sphere Visualization:**
  - Explore the Bloch sphere using online tools and games.
  - Understanding qubit states through interactive visualizations.

  Tools: IBM Quantum Experience, online Bloch sphere simulators.
- **Hands-On Qiskit:**
  - Setting up Python, Jupyter Notebook, and IBM Qiskit.
  - Running one-qubit systems with gates and visualizing outcomes.

  Tools: Qiskit, Jupyter Notebook, IBM Quantum Experience.
- **Linear Algebra Basics:**
  - Vectors and matrices in quantum computing.
  - Vector and matrices addition and multiplication.
  - Apply vector and matrix operations using python and Qiskit.

  Tools: Qiskit, Jupyter Notebook, IBM Quantum Experience.

- **Complex Numbers:**
  - Introduction to complex numbers and the complex plane.
  - Play with Euler's formula in Qiskit for qubit state rotations.

  Tools: Qiskit, Jupyter Notebook, IBM Quantum Experience.

Learning Objectives (LOs):

- Experiment with the Bloch sphere to visualize qubit states.
- Apply linear algebra concepts to basic quantum systems.
- Apply qubit operators by using qiskit to rotate qubits on the Bloch sphere with complex numbers.

Lesson 3: Building Qubits and Gates with Qiskit

Topics:

- **Quantum Gates and Qiskit Basics:**
  - Representing quantum gates in matrix form.
  - Using Qiskit to apply basic single qubit gates and visualize changes on the Bloch sphere: Pauli gates, Hadamard gate, ... , etc.
  - Introduction to quantum circuits and quantum registers.
  - Running quantum circuits in Qiskit.
  - Perform simple quantum operations using Qiskit.
  - Visualize quantum circuits and outcomes.

  Tools: Qiskit, Jupyter Notebook, IBM Quantum Experience.

- **Bra-ket Notation and Normalization:**
  - Introduction through interactive Qiskit experiments.

  Games: Bloch sphere (game from Dimitri) and `Qomposer.app`.

Learning Objectives (LOs):

- Understand and visualize qubit states on the Bloch sphere through Qiskit and games.
- Evaluate mathematical expressions of gate actions on qubits in Qiskit.
- Evaluate the normalization of qubit states using bra-ket notation with real-time feedback in Qiskit.

Lesson 4: Quantum Logic - Multi-Qubit Adventures

Topics:

- **Multiple Qubit Gates:**
  - Explore and simulate SWAP, CNOT, and Toffoli gates.

Tools: Qiskit, Jupyter Notebook, IBM Quantum Experience.

- **superposition:**
  - Introduction to two-qubit systems.
  - Introduce superposition with the action of Hadamard gate.
  - Run a quantum circuit with multiple qubits in Qiskit and check for superposition.

  Tools: Qiskit, Jupyter Notebook, IBM Quantum Experience.

- **Entanglement and Bell States:**
  - Introduce entanglement and Bell states.
  - Run quantum circuits to create Bell states in Qiskit.
  - Interpret and analyze the outcomes of Bell state experiments.

  Tools: Qiskit, Jupyter Notebook, IBM Quantum Experience. **Game:** Quantum Tiq Taq Toe (intermediate level).

Learning Objectives (LOs):

- Simulate multiple qubit systems using Qiskit.
- Apply and analyze multiple quantum gates.
- Evaluate state vector representation of two-qubit system, create quantum superposition, and analyze its effect in the statevector on the qubit states.
- Create and analyze entangled states using Qiskit.

Lesson 5: Real Quantum Computing - Experiments with Quantum Inspire

Topics:

- **Running Quantum Circuits on Real Quantum Computers:**
  - Introduction to Quantum Inspire and its interface.
  - Setting up and running quantum circuits on real quantum hardware.

  Tools: Qiskit, Jupyter Notebook, IBM Quantum Experience.

- **Comparing Simulators and Real Quantum Backends:**
  - Run the same quantum circuits on Qiskit simulators and Quantum Inspire's real quantum backend.
  - Collect and compare results from simulators and real hardware.
  - Analyze differences due to noise and hardware limitations.

  Tools: Qiskit, Jupyter Notebook, IBM Quantum Experience.

- **Applications with Multiple Qubit Systems:**
  - Explore multi-qubit gates such as SWAP, CNOT, and Toffoli on real hardware.
  - Create and analyze entangled states (e.g., Bell states) using real quantum backends.

Tools: Qiskit, Jupyter Notebook, IBM Quantum Experience.

- **Virtual Tour of Quantum Lab, Enschede:**
  - Virtual tour of the Quantum Lab in Enschede by Tjeerd Bollmann.

Learning Objectives (LOs):

- Set up and run quantum circuits on real quantum computers using Quantum Inspire.
- Compare the results of quantum circuits from simulators and real quantum hardware.
- Analyze the impact of noise and hardware limitations on quantum computing results.
- Create and interpret results of multi-qubit systems and entangled states on real quantum backends.

Lesson 6: Quantum Encryption in Action

Topics:

- **Quantum Encryption:**
  - Explore the BB84 protocol through Qiskit simulations.
  - Experiment with encryption applications in quantum systems.
- **Advanced Qiskit: run quantum encryption with Qiskit**
  - Implement quantum encryption techniques like BB84 using Qiskit.

  Tools: Qiskit, Jupyter Notebook, IBM Quantum Experience.
  Game: Cryptris: A quantum encryption game to reinforce the principles of quantum encryption.

Learning Objectives (LOs):

- Implement quantum encryption techniques like BB84 using Qiskit.
- Use short quantum algorithms to explore superposition and entanglement.
- Analyze and interpret the outcomes of quantum encryption simulations.

End of Document