Quantum Computing Foundations with Qiskit

Course Number: QNTM-101

Duration: 3 days

Course Description

This Quantum course teaches technology professionals how to design and execute quantum programs by replacing abstract physics formalism with a visual-first model. Learners build and run real quantum algorithms using Qiskit, translating linear algebra and matrix operations into functional code without first mastering formal quantum mechanics notation. The curriculum introduces foundational quantum effects, including superposition, entanglement, and quantum interference, before progressing to standard algorithms such as Grover's search algorithm and Shor's factoring algorithm. Rather than requiring practitioners to internalize dense physics terminology, the course applies a visual system, Qubelets, to track how individual gates manipulate quantum states, so practitioners can design original circuits rather than reproduce textbook examples. The course extends this foundation into enterprise applications, covering Quantum Key Distribution (QKD), the mechanics of Post-Quantum Cryptography (PQC) migration. Each module closes with a hands-on lab in Qiskit, giving participants direct, verifiable experience building and running circuits rather than passive exposure to concepts.
As federal and enterprise organizations move from quantum-readiness planning toward execution under directives such as National Security Memorandum 10 and OMB Memorandum M-23-02, technical teams are increasingly expected to evaluate quantum hardware, software, and vendor claims with working knowledge of how quantum systems function. This course addresses that gap directly by building quantum-literate staff capable of assessing quantum risk and opportunity from first-hand programming experience rather than secondhand summaries.

Skills Gained

By the end of this course, participants will be able to:

- Design quantum programs from scratch using Qiskit by applying a visual-first model (Qubelets) to track quantum states and gate operations.
- Implement multi-qubit systems and entanglement to evaluate multiple computational paths simultaneously.
- Execute standard quantum algorithms, including Grover's Algorithm for search optimization and Shor's Algorithm for cryptography analysis.
- Assess enterprise vulnerability to quantum attacks and articulate the business drivers for Post-Quantum Cryptography (PQC) migration.
- Map quantum algorithms and post-quantum cryptography standards to specific industry use cases, such as enterprise security and federal compliance.

Who Can Benefit

This course is designed for:

- Software engineers, data scientists, and AI/ML practitioners with prior Python programming experience.
- Technology consultants and IT professionals who need to build practical quantum capabilities.
- Federal agency technical staff and contractors supporting quantum-readiness and PQC migration initiatives.

Prerequisites

Participants should enter this course with:

- Proficiency in Python programming.
- Basic understanding of linear algebra (vectors and matrices).

Organizational Objectives

This course assists organizations to:

- Build practical quantum programming capabilities that transition teams from theoretical awareness to writing and executing Qiskit code.
- Prepare teams for post-quantum cryptography migration by training practitioners on the exact mechanisms of how current encryption breaks.
- Reduce onboarding friction for technical staff by utilizing a visual-first methodology that bypasses traditional physics bottlenecks.
- Establish a baseline quantum literacy across technical teams to enable informed evaluation of quantum hardware and software vendors.
- Build quantum-aware technical staff who can evaluate quantum-resistant algorithms and vendor claims, addressing the technical skills gap that industry surveys cite as a primary barrier to quantum-readiness execution.

Software

All participants require a laptop or desktop computer with a modern web browser and broadband internet.

Course Details

Module 1 – Quantum Literacy and the Qubelets Model

By the end of this module, you will be able to explain how quantum states operate using a visual-first methodology and translate basic quantum gates into executable matrix operations. You will also be able to distinguish between classical bits and qubits without relying on abstract physics notation.

- From classical bits to qubits
- The visual-first approach: green pentagons (|0?) and red triangles (|1?)
- Visualizing amplitudes and phases through shape orientation
- Quantum logic gates for Boolean operations
- Tracking quantum interference through visual cancellation
- Hands-on Lab: Build and execute a basic quantum circuit in Qiskit using the Qubelets mental model to predict and verify the output.

Module 2 – Quantum Effects and Algorithm Design

By the end of this module, you will be able to implement multi-qubit systems and apply standard quantum algorithms to solve specific computational problems. You will also be able to trace how quantum parallelism evaluates multiple states simultaneously.

- Introducing superposition and tagging gates
- Multi-qubit systems and the "mega-qubit" concept
- Entanglement and cross-qubit interactions
- Shor's Algorithm and number theory foundations for cryptography
- Hands-on Lab: Build a quantum tagging circuit to isolate a specific state, tracing how interference and phase manipulation collapse the mega-qubit into a targeted outcome.

Module 3 – Post-Quantum Cryptography and Enterprise Applications

By the end of this module, you will be able to assess enterprise vulnerability to "harvest now, decrypt later" quantum threats using Mosca's Theorem and map NIST's finalized PQC standards to enterprise cryptographic assets. You will also be able to draft a hybrid, crypto-agile migration timeline aligned to current federal deadlines.

- The "harvest now, decrypt later" (HNDL) threat and Mosca's Theorem
- Quantum Key Distribution (QKD) vs. Post-Quantum Cryptography (PQC)

- NIST's finalized standards: ML-KEM (FIPS 203, formerly CRYSTALS-Kyber), ML-DSA (FIPS 204, formerly CRYSTALS-Dilithium), and SLH-DSA (FIPS 205, formerly SPHINCS+)
- Hybrid architectures and crypto-agile migration strategies
- NSM-10 and OMB M-23-02: federal cryptographic inventory and migration planning
- The June 2026 Executive Order and NSA CNSA 2.0: concrete 2030 (key establishment) and 2031 (signatures) deadlines
- Hands-on Lab: Using a mock enterprise PKI/TLS asset inventory, assess Harvest-Now-Decrypt-Later risk with Mosca's Theorem, select appropriate ML-KEM and ML-DSA parameter sets for each asset, and draft a phased hybrid migration timeline that verifies compliance against the 2030 and 2031 federal deadlines.