Mastering Qiskit v2.0 – From Fundamentals to Hardware
Phase #1

Instructor: Ricard Santiago Raigada García

Topic Area:	Quantum Computing, Qiskit SDK 2.0, IBM Quantum Hardware
Date of Teaching:	August 7, 2025
Time Allotted:	1 Month
Level:	Graduate – Introductory to Intermediate (Advanced Theoretical Quantum Mechanics Prerequisite)
Delivery Mode:	Hands-on Lecture with Complementary Theory
Associated Materials:	Official repository of the Mastering Qiskit v2.0 course

Contents

1 Introduction

The *Mastering Qiskit v2.0* course is a structured learning pathway for individuals seeking a practical foundation in quantum computing using Qiskit v2.0. While the course is independent of IBM, it is aligned with the current IBM Quantum technology stack to ensure learners acquire skills directly applicable to real hardware workflows.

This program is the result of my combined background in computer engineering, data science, and ongoing studies in theoretical and computational physics.

2 Learning Outcomes

Upon completing Phase 1 of the program, learners will be able to:

- Configure and connect to IBM Quantum hardware environments.
- Prepare and manipulate quantum states using Qiskit primitives.
- Apply common circuit construction patterns and library components.
- Implement diverse measurement strategies and interpret results.
- Explore and apply fractional gates and advanced circuit techniques.
- Work with operators, observables, and optimize execution on real devices.
- Execute jobs on IBM Quantum hardware and learn error mitigation methods.

3 Phase 1 Curriculum – Detailed Programme

This phase comprises 22 episodes designed to progressively build proficiency in quantum computing using Qiskit v2.0. Each episode includes a detailed description, specific learning objectives, and the competencies that learners will acquire upon completion. The sequence includes the following episodes:

3.1 Fundamentals

Episode 1 – Python Environment Setup

Description: Introduces the fundamental tools and environment configuration required to work with Qiskit v2.0. Learners will install Python, configure virtual environments, and prepare their systems for quantum computing development.

Objectives:

1. Prepare and configure Python 3.10 or higher and relevant package managers.
2. Create and activate virtual environments for project isolation.
3. Install Qiskit v2.0 and verify correct setup.

Competencies Acquired:

- Ability to manage Python environments effectively.
- Installation and basic configuration of Qiskit.
- Understanding of dependency management for reproducible workflows.

Episode 2 – Connect to IBM Quantum with IBM Cloud

Description: Guides the learner through the process of creating an IBM Quantum account, generating API tokens, and connecting to IBM Quantum services from the Qiskit SDK.

Objectives:

1. Register and set up an IBM Quantum account.
2. Obtain and securely store an API token.
3. Establish a connection to IBM Quantum using Qiskit.

Competencies Acquired:

- Authentication and secure credential management.
- Connecting local environments to cloud-based quantum resources.
- Basic navigation of the IBM Quantum platform.

Episode 3 – Preparing Quantum States – One Qubit

Description: Covers fundamental single-qubit state preparation using Qiskit's circuit model. Introduces statevector visualisation and basic gate operations.

Objectives:

1. Construct single-qubit circuits using Qiskit.
2. Apply Pauli-X, Y, Z, and Hadamard gates.
3. Visualise quantum states on the Bloch sphere.

Competencies Acquired:

- Single-qubit quantum state manipulation.
- Understanding of basic quantum gates.
- Visual interpretation of quantum state evolution.

Episode 4 – Preparing Quantum States – Superposition

Description: Explores creating and measuring superposition states, including uniform superposition with Hadamard transforms.

Objectives:

1. Apply Hadamard gates to create superposition.
2. Interpret measurement outcomes of superposed states.
3. Simulate circuits with Qiskit Aer and analyse results.

Competencies Acquired:

- Superposition state preparation.
- Quantum probability interpretation.
- Proficiency with Qiskit simulators.

Episode 5 – Preparing Multi-Qubit States – Bell States and Ordering

Description: Introduces two-qubit entanglement through the preparation of Bell states and explains qubit ordering conventions in Qiskit.

Objectives:

1. Construct Bell states using H and CNOT gates.
2. Differentiate between big-endian and little-endian ordering in Qiskit.
3. Measure and verify entanglement correlations.

Competencies Acquired:

- Understanding of entanglement.
- Mastery of two-qubit gate sequences.
- Awareness of qubit ordering conventions in Qiskit.

3.2 Qiskit Foundations

Episode 6 – Introduction to the Qiskit Stack

Description: Provides an overview of Qiskit's architecture —Qiskit SDK, Qiskit Runtime, and additional services.

Objectives:

1. Identify the main components of the Qiskit stack.
2. Understand the role of each module in quantum workflows.
3. Explore documentation and API references.

Competencies Acquired:

- Familiarity with Qiskit's modular structure.
- Ability to choose the right Qiskit component for a given task.
- Efficient use of Qiskit documentation.

Episode 7 – Introduction to Qiskit Patterns

Description: Introduces reusable design patterns for building quantum circuits, reducing code redundancy and improving readability.

Objectives:

1. Recognise common Qiskit coding patterns.
2. Implement circuit templates for repeated structures.
3. Apply patterns to improve maintainability.

Competencies Acquired:

- Application of reusable coding structures.
- Increased productivity in circuit design.
- Improved code maintainability.

Episode 8 – Using the Qiskit Circuit Library

Description: Explores Qiskit's pre-built circuit library for rapid prototyping and testing of quantum algorithms.

Objectives:

1. Access and import circuits from Qiskit's circuit library.
2. Customise predefined circuits with parameters.
3. Integrate library components into larger workflows.

Competencies Acquired:

- Rapid development using pre-built circuits.
- Ability to adapt templates for specific use cases.
- Familiarity with Qiskit's reusable resources.

Episode 9 – Constructing Circuits Programmatically

Description: Teaches how to dynamically construct quantum circuits using Python logic, loops, and conditional statements.

Objectives:

1. Use loops to generate scalable circuit structures.
2. Implement parameterised circuits.
3. Combine classical control with quantum operations.

Competencies Acquired:

- Programmatic circuit generation.
- Efficient scaling for multi-qubit architectures.
- Classical-quantum hybrid logic implementation.

3.3 Measurement and Gates

Episode 10 – Measuring Qubits

Description: Covers measurement operations in Qiskit, from single-qubit measurement to full register readout.

Objectives:

1. Add measurement operations to quantum circuits.
2. Map quantum measurements to classical bits.
3. Interpret histogram results from simulations.

Competencies Acquired:

- Mastery of measurement syntax in Qiskit.
- Understanding quantum-to-classical mapping.
- Statistical analysis of measurement outcomes.

Episode 11 – Fractional Gates

Description: Introduces rotation gates with arbitrary angles, explaining their relationship to standard quantum gates and their applications in fine-grained state control.

Objectives:

1. Use R_x, R_y, and R_z gates with parameterised angles.
2. Relate fractional gates to standard gates via special angles.
3. Apply fractional rotations to implement specific quantum transformations.

Competencies Acquired:

- Proficiency with parameterised single-qubit rotations.
- Understanding of gate equivalences and decompositions.
- Fine control over quantum state manipulation.

3.4 Advanced Circuit Techniques

Episode 12 – Implementing Classical Feedforward Logic and Conditional Execution

Description: Demonstrates how to integrate classical measurement results into conditional quantum operations, enabling feedforward control within circuits.

Objectives:

1. Implement classical conditionals using measurement results.
2. Apply quantum gates conditionally based on classical bit values.
3. Design algorithms requiring adaptive behaviour.

Competencies Acquired:

- Hybrid classical-quantum programming skills.
- Understanding of conditional execution in Qiskit.
- Ability to design adaptive quantum algorithms.

Episode 13 – Understanding Qubit and Bit Ordering in Qiskit

Description: Explains Qiskit's qubit indexing conventions and how they affect circuit design, measurement ordering, and result interpretation.

Objectives:

1. Understand Qiskit's default little-endian ordering.
2. Reorder qubits and classical bits to match algorithmic needs.
3. Interpret output bitstrings correctly in different orderings.

Competencies Acquired:

- Confidence in handling bit and qubit ordering.
- Avoiding common indexing mistakes in multi-qubit systems.
- Improved debugging and interpretation of circuit results.

Episode 14 – Controlling Timing and Stretch for Quantum Scheduling

Description: Introduces circuit scheduling concepts, including gate timing, delay insertion, and pulse stretching for hardware control.

Objectives:

1. Use delay instructions to control gate execution timing.
2. Stretch pulses for hardware-specific needs.
3. Apply scheduling strategies to optimise performance.

Competencies Acquired:

- Understanding of quantum circuit timing control.
- Ability to synchronise operations in multi-qubit systems.
- Basic awareness of pulse-level hardware constraints.

Episode 15 – Re-Synthesising Unitary Operators for Circuit Optimization

Description: Explains how to re-synthesise unitary operators to reduce gate counts and circuit depth without altering functionality.

Objectives:

1. Decompose arbitrary unitaries into efficient gate sequences.
2. Apply Qiskit's synthesis passes to optimise circuits.
3. Compare original and optimised circuits for performance gains.

Competencies Acquired:

- Skills in circuit optimisation.
- Understanding of unitary synthesis and decomposition.
- Ability to balance circuit depth vs. fidelity trade-offs.

Episode 16 – Saving and Loading Circuits with QPY Serialization

Description: Teaches how to persist and reload quantum circuits using QPY format, ensuring exact reconstruction across environments.

Objectives:

1. Save Qiskit circuits to QPY files.
2. Load QPY files into active sessions.
3. Verify circuit integrity post-loading.

Competencies Acquired:

- Ability to store and share quantum circuits efficiently.
- Skills in reproducibility and collaboration.
- Familiarity with Qiskit's serialisation tools.

3.5 Operators and Execution

Episode 17 – Exploring Operator Classes and Their Applications

Description: Introduces Qiskit's operator classes for representing quantum operators, observables, and their algebraic properties.

Objectives:

1. Create and manipulate operators in Qiskit.
2. Perform basic arithmetic and tensor operations.
3. Apply operators as observables in measurements.

Competencies Acquired:

- Understanding operator algebra in quantum mechanics.
- Ability to define custom observables.
- Integration of operators into algorithm workflows.

Episode 18 – Measuring Observables in the Pauli Basis

Description: Demonstrates how to measure quantum states in the Pauli-X, Pauli-Y, and Pauli-Z bases.

Objectives:

1. Apply basis-change gates before measurement.
2. Measure in different Pauli bases.
3. Interpret results from non-computational basis measurements.

Competencies Acquired:

- Mastery of basis transformations.
- Skills in extracting specific observables from quantum states.
- Practical understanding of Pauli measurements.

Episode 19 – Working with the Operator Class in Depth

Description: Expands on Episode 17 by exploring advanced operator manipulations, including commutators, eigen-decompositions, and operator exponentiation.

Objectives:

1. Compute commutators and anti-commutators.
2. Perform spectral decompositions of operators.
3. Exponentiate operators for time-evolution simulation.

Competencies Acquired:

- Deep understanding of quantum operator mathematics.
- Capability to simulate Hamiltonian dynamics.
- Analytical thinking for algorithm design.

Episode 20 – Hardware-Aware Circuit Optimization with the Qiskit Transpiler

Description: Shows how to use the transpiler to optimise circuits for specific IBM Quantum hardware constraints and coupling maps.

Objectives:

1. Configure transpiler passes for target backends.
2. Reduce SWAP gates via coupling map awareness.
3. Balance transpilation optimisation levels with execution fidelity.

Competencies Acquired:

- Hardware-aware algorithm deployment.
- Skills in backend-specific circuit optimisation.
- Proficiency with transpiler configuration.

Episode 21 – Running Quantum Jobs on IBM Hardware with Qiskit Runtime

Description: Guides the learner through executing circuits on real IBM Quantum devices using Qiskit Runtime services.

Objectives:

1. Submit jobs to IBM Quantum hardware.
2. Monitor and retrieve job results.
3. Apply runtime configurations for performance gains.

Competencies Acquired:

- Practical experience with hardware execution.
- Familiarity with runtime optimisation parameters.
- Skills in job monitoring and result management.

Episode 22 – Analyzing Results and Applying Error Mitigation

Description: Introduces error mitigation strategies for improving result quality when running circuits on noisy quantum hardware.

Objectives:

1. Analyse raw hardware output.
2. Apply measurement error mitigation techniques.
3. Evaluate and compare mitigated vs. unmitigated results.

Competencies Acquired:

- Awareness of quantum noise sources.
- Ability to apply post-processing mitigation methods.
- Critical evaluation of quantum experiment accuracy.

4 Pedagogical Approach

The course follows a modular and incremental approach:

- Each episode focuses on a specific technical skill or concept.
- Concepts are reinforced through exercises, challenges, and real hardware execution examples.
- Practical coding demonstrations are prioritized to bridge theory and implementation.

5 Future Work

Phase 2 will expand into advanced topics such as:

- Deep understandig on pass managers.
- Profound understandig on transpilation.

6 Acknowledgements

Special thanks to the Qiskit open-source community for fostering an inclusive and collaborative environment that inspires projects like this one.

7 References

- Qiskit Documentation: `https://docs.quantum.ibm.com/`
- Mastering Qiskit v2.0 – Zenodo DOI: `https://doi.org/10.5281/zenodo.16872377`