# https://www.nobleprog.in/quantum-computing-training

# 1. Quantum Computing with IBM Quantum Experience Training Course

Course Code

quantumibm

Duration

14 hours (usually 2 days including breaks)

Requirements

* An understanding of Python programming

**Audience**

* Computer scientists

Overview

Quantum computing is the advancement of the binary system, leveraging quantum phenomena to record information. IBM Quantum Experience is an open source framework that is used to create, edit, and invoke quantum circuits in an interface like simulation.

This instructor-led, live training (online or onsite) is aimed at computer scientists who wish to create, edit, and invoke a quantum circuit using IBM Quantum Experience.

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

* Set up the necessary development environment to start creating quantum circuits.
* Improve on inaccuracies in quibits using IBM Quantum Experience and its simulations.
* Control gate behavior using the gate's native counterpart.
* Schedule the time of gate events.

**Format of the Course**

* Interactive lecture and discussion.
* Lots of exercises and practice.
* Hands-on implementation in a live-lab environment.

**Course Customization Options**

* To request a customized training for this course, please contact us to arrange.

Course Outline

Introduction

Quantum Information Theory

* The uncertainty theory
* Superimposition and entanglement
* Subatomic particles

Overview of Classic Computers

* Bits
* Binary systems
* Transistors

Quantum Computing

* Topological codes
* Circuit QED
* Quibits

IBM Quantum Experiences and Python Frameworks

* Modules and processes used in quantum computing
* Quantum circuit simulations
* Circuit composer
* Qiskit Notebooks

Preparing the Development Environment

* Installing and configuring Anaconda
* Installing and configuring pip and Python packages
* Installing Qiskit
* Configuring IBM Quantum Experience
* Configuring the IDE as the project manager
* Setting up a version control center (GitHub)

Building QER Circuits in Circuit Composer

* Applying gates
* Measuring quibits
* Saving and running the QER Circuits

Coding QER Circuits in Qiskit Notebooks

* Implementing a noisy simulation
* Coding a noisy circuit
* Implementing a quantum circuit for Quantum Fourier Transform on 4 quibits
* Creating a basic algorithm in a quantum circuit
* Coding a quantum teleport circuit

Testing the Cirquits

* Running the QER circuits on a real IBM quantum device
* Analyzing the displayed outcomes

Summary and Conclusion

# 2. Fundamentals of Quantum Computing and Quantum Physics Training Course

Course Code

quantum

Duration

21 hours (usually 3 days including breaks)

Requirements

* Knowledge of mathematical methods in probability and linear algebra
* Comprehension of foundational computer science theories and algorithms
* An understanding of elementary quantum physics concepts
* Basic experience with quantum mechanics models and theories

**Audience**

* Computer Scientists
* Engineers

Overview

Quantum computing is the integration of quantum physics, mathematics, and computer science methods for the advancement of computational models. It applies two main quantum properties namely superposition and entanglement, which allows the development of quantum computers. Quantum computing incorporates these behaviors of quantum particles to execute computational technologies that are exponentially faster than classic computers.

This instructor-led, live training (online or onsite) is aimed at computer scientists and engineers who wish to understand the principles behind quantum computing and utilize them in developing algorithms for quantum computer implementations.

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

* Comprehend the fundamentals of quantum computing.
* Understand and apply quantum physics concepts into computational methods.
* Create algorithms for quantum computers.
* Solve computational problems efficiently with quantum computers.
* Integrate quantum behaviors into existing computational models.
* Perceive the potential of quantum computing in the advancement of other technologies.

**Format of the Course**

* Interactive lecture and discussion.
* Lots of exercises and practice.
* Hands-on implementation in a live-lab environment.

**Course Customization Options**

* To request a customized training for this course, please contact us to arrange.

Course Outline

Introduction

Overview of Quantum Physics Theories Applied in Quantum Computing

* Fundamentals of quantum superposition
* Fundamentals of quantum entanglement
* Mathematical foundations of quantum computing

Overview of Quantum Computing

* Differentiating quantum computing and classical electronic computing
* Integrating quantum behaviors into quantum computing
* The Qubit
* Implementing the Dirac notation
* Computational basis measurements in quantum computing
* Quantum circuits and quantum oracles

Working with Vectors and Matrices in Quantum Computing

* Matrix multiplication using quantum physics
* Conventions of tensor products

Applying Advanced Matrix Concepts to Quantum Computing

Overview of Quantum Computers and Quantum Simulators

* The quantum hardware and its components
* Running a quantum simulator
* Executable quantum mechanisms in a quantum simulation
* Performing quantum computations in a quantum computer

Working with Quantum Computing Models

* Logic and functions of different quantum gates
* Understanding superposition and entanglement effects on quantum gates

Utilizing Shor's Algorithm and  Quantum Computing Cryptography

Implementing Grover's Algorithm in Quantum Computing

Estimating a Quantum Phase in a Quantum Computer

* The quantum Fourier transform

Writing Basic Quantum Computing Algorithms and Programs for a Quantum Computer

* Utilizing the right tools and language for quantum computing
* Setting up quantum circuits and specifying quantum gates

Compiling and Running Quantum Algorithms and Programs in a Quantum Computer

Testing and Debugging Quantum Algorithms and Quantum Computer Programs

Identifying and Correcting Algorithm Errors Using Quantum Error Correction (QEC)

Overview of Quantum Computing Hardware and Architecture

Integrating Quantum Algorithms and Programs with the Quantum Hardware

Troubleshooting

Advancing Quantum Computing for Future Quantum Information Science Applications

Summary and Conclusion

# 3. Practical Quantum Computing Training Course

## Course Code

quantum001

## Duration

10 hours (usually 2 days including breaks)

## Requirements

No background in quantum computing, or quantum physics necessary.

No physics background necessary.

We cover A to Z of Quantum Computing!

## Overview

### Practical Quantum Computing: Live Online

### Launch your high-tech career

This is a 10 hour instructor-led, live online training course. After your immersive training, you will be ready to start work as an entry level quantum computing developer.

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

* **Run and test your quantum programs with the integrated IBM Q**
* **Use Qiskit to create, compile, and execute quantum computing programs**
* **Working with practical and advanced quantum algorithms such as QAOA**
* **Recast real-world problems into an appropriate** **quantum computing language**

**Format of the Course**

* Interactive lecture and discussion.
* Lots of exercises and practice.
* Hands-on implementation in a live-lab environment.

**Course Customization Options**

* To request a customized training for this course, please contact us to arrange.

## Course Outline

* Basic notions of Quantum Mechanics
* Introduction to Quantum Computing
* Quantum Gates and Quantum Circuits (Binary Quantum Gates)
* Quantum Computation via Python and Qiskit
* Practical Quantum Algorithm Design and Construction
* Advanced Quantum Algorithm Implementations via Qiskit
* Solving real world problems in diverse industries via IBM’s quantum computers

# 4. Quantum Computing with Cirq Framework Training Course

Course Code

cirq

Duration

21 hours (usually 3 days including breaks)

Requirements

* An understanding of software and hardware computing concepts.
* An understanding of complex numbers and linear algebra.

**Audience**

* Developers
* Scientists
* Engineers

Overview

Quantum computers are machines that are built on the principles of quantum mechanics. Unlike classical computers which rely on bits that can only exist in one of two states (0 or 1), quantum computers use quantum bits which can exist in multiple states at once. Quantum computers are expected to process algorithms and solve problems much faster than classical computers.

Cirq is a Python library for writing, manipulating, and optimizing quantum circuits and running them against quantum computers and simulators.

In this instructor-led, live training (onsite or remote), participants will learn the fundamentals of quantum computing as they step through the development of algorithms targeting quantum computers.

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

* Set up the necessary development environment to start creating quantum circuits.
* Control gate behavior using the gate's native counterpart.
* Improve on quibit inaccuracies using Cirq.
* Write and execute a quantum circuit to run against a NISQ based quantum computer simulator.

**Format of the Course**

* Interactive lecture and discussion.
* Lots of exercises and practice.
* Hands-on implementation in a live-lab environment.

**Course Customization Options**

* To request a customized training for this course, please contact us to arrange.
* To learn more about Ciq Framework, please visit: https://github.com/quantumlib/Cirq

Course Outline

Introduction

Quantum Information Theory

* The uncertainty theory
* Superposition and entanglement
* Subatomic particles

Overview of Classic Computers

* Bits
* Binary systems
* Transistors

Quantum Computing

* Topological codes
* Circuit QED
* Quibits

Understanding Quantum Computers and NISQ Processors

* Algorithms for the NISQ Architecture

Overview of Cirq Framework

Overview of Quantum Development Kits

Setting Up the Quantum Development Environment

Cirq Data Structures and Syntax

Preparing the Quantum Machine Simulator

Case Study: Low Depth Quantum Algorithms for Quantum Chemistry Problems

Writing a Quantum Algorithm

Controlling Quantum Circuits

Specifying Gate Behavior and Placement

Scheduling the Timing of Gates

Compiling the Algorithm

Dealing with the Constraints of Quantum Hardware

Running the Algorithm on a Local Simulator

Testing and Debugging an Algorithm

Integration with Cloud Simulators

Integration with Future Quantum Hardware

Troubleshooting

The Future of Quantum Computing  
   
Summary and Conclusion

# 5. Getting Started with Quantum Computing and Q# Training Course

Course Code

quantumcomputingqsharp

Duration

14 hours (usually 2 days including breaks)

Requirements

* Experience with Visual Studio

Overview

Quantum computers are machines that are built on the principles of quantum mechanics. Unlike classical computers which rely on bits that can only exist in one of two states (0 or 1), quantum computers use quantum bits which can exist in multiple states at once. Because of this, quantum computers are expected to process algorithms and solve problems much faster than classical computers. Q# is a programming language created by Microsoft specifically for quantum computing.

In this instructor-led, live training, participants will learn the fundamentals of quantum computing and Q# as they step through the development of simple quantum programs.

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

* Install and configure Microsoft's Quantum Development Kit.
* Understand the concepts behind quantum computing.
* Build, test, execute, and troubleshoot a quantum program using Q#, Visual Studio and a local quantum computing simulator.

**Audience**

* Developers

**Format of the course**

* Part lecture, part discussion, exercises and heavy hands-on practice

**Note**

* To request a customized training for this course, please contact us to arrange.

Course Outline

Introduction to Quantum Computing

* Quantum Computing vs. Classical Computing

Overview of Microsoft's Quantum Development Kit

* Q# Language and Compiler
* Q# Standard Library
* Local Quantum Machine Simulator
* Quantum Trace Simulator
* Visual Studio Extension

Setting Up the Quantum Development Environment

* Installing and Configuring Microsoft's Quantum Development Kit

Using the Q# Programming Language

* The Type Model
* Expressions
* Statements
* File Structures
* Operations and Functions
* Working with Qubits

Creating Your First Quantum Program with Q#

* Setting Up a Quantum Solution and Project in Visual Studio
* Defining Q# Operations
* Calling Q# Operations using C#

Executing Your Quantum Program

* Using the Local Quantum Computing Simulator

Testing and Debugging Your Quantum Program

Troubleshooting

Summary and Conclusion