**Quantum Computing with Black Opal:**

| | |
|---|---|
| **Course Objectives** | • Deliver the fundamentals of quantum computing, from key concepts through to programming a quantum computer. Students will be able to explain computational complexity arguments for why certain algorithms can deliver quantum advantage.<br>• Enable students to identify the practical, real-world applications of quantum computing to exhibit relevance in various industries and current state-of-the-art knowledge of applicability.<br>• Empower a student to recognize and successfully use key terminology relevant to quantum computing hardware and software - qubits, gates, circuits, algorithms, coding formats, etc<br>• Deliver physical intuition for key underlying concepts that lead to the differentiation of quantum computing including the superposition principle, qubits, measurement and entanglement<br>• Enable a student to build prototype quantum algorithms by recognizing the fundamental logical organization of a quantum circuit, and demonstrate the key modules needed to create the programs to be run on quantum computers<br>• Enable student to identify the key programming abstractions in a quantum computer, identify quantum intermediate representations and link concepts of quantum circuits and states to commands |
| **Course Outcomes** | • Create quantum circuits to be run on quantum computers, and be able to interpret the circuit diagrams used in the field<br>• Build foundational quantum algorithms with real word applications such as Grover's Search and QAOA and program a quantum computer using common programming languages<br>• Identify and articulate the technological challenges in developing useful quantum computers and apply this knowledge in a business context and pathway to relevance in various industries (noise, error, size, etc). |

**Course Duration:** 45 Hours

**Course Content:**

## Unit 1: Introduction to Quantum Computing and Superposition

- Introduction to quantum computing:
  - What is quantum computing?: Discover the essence of quantum computers and why we care about building them.
  - How do quantum computers work?: Quantum computers work according to different rules. Understand the essence of those rules and what powers they enable.
  - Will quantum computers break the internet?: Quantum computers can break the codes that are used for security online. Find out exactly how they do this, and how we plan to mitigate the risk.
  - How to build a quantum computer: Quantum computing isn't just theoretical. Discover how researchers and companies are building real systems right now!
  - Analogy toolbox: Quantum computing is phrased in a technical language. Here is a simple toolbox of analogies to apply when the jargon starts to fly.
- Superposition:
  - Develop an intuitive understanding of superposition as interference of waves.
  - Gain an abstract representation of superposition in the polarization degree of freedom possessed by waves.
  - Superposition in example quantum systems relevant to quantum computing: ions, atoms, superconducting circuits and photons.
  - Superposition of abstract objects representing quantum systems.
  - Superposition in a quantum computation and quantum parallelism.

## Unit 2: Qubits and Measurement

- Qubits:
  - It from bit: introduces the concept of information. Learn how information is defined and represented, as well as how it is measured and used.
  - Qubit from bit: introduces quantum information. Learn how quantum information is written and read and how it is different from digital information.
  - The Bloch circle: a first step towards a visualization of qubits: a circle. Learn how a qubit is written, read, and processed in a geometric picture.
  - Do you |ket⟩ it?: Learn about the abstract representation of qubits, what these objects mean and how to use them.
  - The Bloch sphere: Interact directly with the Bloch sphere to represent the full set of qubit states.

- Measurement:
  - Bloch party: The concept of quantum measurement using the Bloch sphere - a visual representation of a qubit of information - is introduced. Explore measurement through interacting with a qubit.
  - Randomness rules: Why are quantum measurements random and what rules govern them? Understand how the symbolic notation of kets translates into real measurement outcomes and discover the Born rule.
  - Get real: Measurement in real quantum systems relevant to quantum computing: trapped ions, atoms, superconducting circuits, and more.
  - What's the problem: What is the measurement problem and is it really a problem? A brief detour in metaphysics at one end of the spectrum and cold hard calculation at the other.

**Unit 3: Circuits and Entanglement**

- Circuits:
  - Single qubit warm-up: A quantum circuit is a useful tool to visualize and create quantum instructions or algorithms. We will begin with circuits for a single qubit.
  - Double the fun: Two-qubit gates open a whole new world of quantum possibilities, and are necessary to harness the power of quantum computation.
  - More, more, more! A real quantum algorithm will involve many qubits. In this topic, we will dial the qubit numbers up… to three at least.
  - You, quantum mechanic: A complete quantum toolbox consists of classical gates and quantum gates. Get familiar with the quantum toolbox and know which is best suited for the job at hand.
  - Quantum search: Strap in — we're going full quantum. Use a quantum algorithm to search a database!
- Entanglement:
  - Bird's eye view: The concept of quantum entanglement is considered the most complicated and mysterious topic in all of science. Get a picture of why it's important and how it's used.
  - Correlations and kets: Entanglement can be made precise using ket notation and qubits. Here students learn to recognize which qubit states are entangled and why.
  - Creating entanglement: Learn how to create and visualize entanglement with a few simple instructions.
  - Entanglement in the wild: Generate entanglement in real quantum systems relevant to quantum computing: ions, atoms, superconducting circuits, and more.

  - Variables in hiding: The concept of quantum entanglement is considered the most complicated and mysterious topic in all of science. Get a picture of why it's important and how it's used.
  - Visualizing entanglement in quantum computers: Learn how we can use a familiar friend - the Bloch sphere - to visualize what happens when qubits become entangled and just how much things change.

**Unit 4: Noise and Control**

- Noise:
  - NISQy business: We have noisy quantum computers now, and may have them for some time. Are they useless, or can something interesting be done with them?
  - Bloching noise: What does the effect of noise look like on qubits? See what noise does to qubit states with the Bloch sphere.
  - Controlling noise: How can noise be mitigated? How can instructions be implemented in the presence of noise? Learn to get it under control.
  - Error correction: Remove errors altogether with error correction.
- Control:
  - Paradigm shifts: What is control? Where is it used and why is it helpful? Some insights from the general theory of control can give us intuition over the quantum problem.
  - Making gates: We know how powerful control can be. In reality, quantum logic gates are physical operations that take time. Here students will design gates using quantum control.
  - Noise be gone: Here students will learn about quantum control and how it is used to design quantum gates and algorithms that overcome the fundamental challenge in the field - error.
  - The big show: How is quantum control used in practice? It is used in every step of the process of solving the problem of creating quantum technology. Here students walk through that process.

**Unit 5: Speedup and Programming**

- Speedup:
  - What is complexity?: Proving one algorithm is better than another is not as easy as one might think. Understand some of the key jargon encountered in debates about the utility of quantum computers.
  - Deutsch-Jozsa: The first quantum algorithm displaying a speedup was dramatic and illustrative. A deep dive into the Deutsch-Jozsa algorithm.
  - Shor's algorithm: Perhaps the most famous algorithm is the factoring algorithm due to Peter Shor, which provides a speedup in the task of factoring large numbers.

- Grover speedups: non-exponential, yet still significant speedups are known for search and related problems. These might be the first useful applications of quantum computing.

- Programming:
  - Gate, set, match: Students are introduced to a set of gates, and must understand how to actually make them operate.
  - Compiling: Abstract circuits cannot be directly mapped to real quantum computers. Compiling is the process of translating one quantum program to another, and is how we get quantum computers to calculate.
  - Code: Quantum programming languages allow the rapid development of software to run problem-solving algorithms on quantum computers. Students start with the basics of assembly-level programming.
  - Quantum optimization: Quantum computers today are probably running an algorithm with an acronymic name like QAOA. What are these and why do they work with noisy quantum computers? Here, students will find out.

**Test Projects:**

**Use Cases**

| **S.No.** | **Topic** | **USE CASES** |
|---|---|---|
| **1** | Introduction: Discover the essence of quantum computers and why we care about building them. | **Use case:** Evaluate the general applicability of quantum computers to a given problem type<br>**Scenario**: An engineer is<br>called upon to evaluate the future impact of quantum computing on their industry<br>**Task:** The engineer is able to use their broad understanding of the methods, timescales and problem types that quantum computers are developing to assess how a business may wish to invest or get involved in quantum computing |
| **2** | Introduction: Quantum Computers work according to different rules. Understand the essence of those rules | **Use Case**: Evaluate whether a classical or quantum solution is going to be applicable |

| | | |
|---|---|---|
| | and what powers they enable. | **Scenario:** Given a specific problem type, determine whether quantum computing is applicable or beneficial<br>**Task:** Knowing how a quantum computing encodes and solves a problem, the engineer is able to determine whether quantum computing is a good fit for a given problem |
| **3** | Introduction: Quantum computers can break the codes that are used for security online. Find out exactly how they do this, and how we plan to mitigate the risk. | **Use case:** Upgrade an organisation to use post-quantum cryptography<br>**Scenario:** An information security officer is called upon to review IT security policies for their company or organization<br>**Task:** The officer uses their knowledge of cryptography vulnerabilities and quantum computers to assess their own organization's policies and goals, and design a suitable security policy |
| **4** | How to build a quantum computer: Quantum computing isn't just theoretical. Discover how researchers and companies are building real systems right now! | **Use case:** Assist in the selection of a specific type of quantum hardware<br>**Scenario:** An engineer is asked to review the state of quantum computing to inform how an organisation might start to get involved in quantum computing<br>**Task:** Knowing the various pros and cons of an evolving hardware space, the engineer is able to appreciate the variety of options available, track the developing industry and decide on specific hardware as appropriate |
| **5** | Analogy toolbox: Quantum computing is phrased in a technical language. Here is a simple toolbox of analogies | **Use case:** Know the key terms to participate in quantum-related business decisions<br>**Scenario:** An analyst is asked to get involved |

| | | |
|---|---|---|
| | to apply when the jargon starts to fly. | with the quantum initiative at a business<br>**Task:** Because the analyst is familiar with the most common terms used in quantum, they are able to meaningfully participate in business decisions relating to quantum computing |
| **6** | Superposition:<br>Develop an intuitive understanding of superposition as interference of waves. | **Use case:** Learn about the fundamental characteristics of the superposition principle<br>**Scenario:** Abstract waves are represented visually in 1D and 2D in an interactive interface.<br>**Task:** Students will use interactive controls to explore superpositions of 1D and 2D waves and will appropriately tune key parameters including amplitude, wavelength, and phase. |
| **7** | Superposition: Gain an Abstract representation of superposition in the polarization degree of freedom possessed by waves. | **Use case:** Explore the superposition principle through a visual representation<br>**Scenario**: Waves are linked to mechanical excitations of a jump rope and the properties of photons.<br>**Task:** Students will use interactive controls to manipulate properties of the mechanical excitation in order to correctly apply concepts of polarization. They will then extend these insights to the properties of light in order to link wave excitation properties to notions of linear and circular polarization. |
| **8** | Superposition in example quantum systems relevant to quantum computing: ions, atoms, superconducting circuits and photons. | **Use case:** Create superposition on a quantum computer using a specific architecture<br>**Scenario:** An engineer is asked to create superposition on a quantum computer using superconducting qubits |

| | | |
|---|---|---|
| | | **Task:** By understanding the specifics of how superposition takes place taken on a quantum computer using superconducting qubits, the engineer is able to better interpret the results of executing a circuit on that architecture |
| **9** | Superposition of abstract objects representing quantum systems. | **Use case:** Communicate superposition<br>**Scenario:** Two engineers need to be able to mathematically describe superposition<br>**Task:** The engineer is able to use his knowledge of kets to describe superposition using universally accepted notation, such as in research literature |
| **10** | Superposition in quantum computation and quantum parallelism. | **Use case:** design a Quantum Key Distribution (QKD) protocol<br>**Scenario:** A communications engineer must calculate bandwidth requirements for a QKD system<br>**Task:** The engineer draws on a thorough understanding of superposition in quantum systems to inform the design process |
| **11** | Qubits: introduces the concept of information. Learn how information is defined and represented, as well as to how it is measured and used. | **Use case:** Assemble a digital circuit<br>**Scenario:** Assemble a circuit composed of digital logic gates<br>**Task:** Understanding the operations of both digital logic gates and quantum gates is essential for an engineer to competently create a quantum circuit |
| **12** | introduces quantum information. Learn how quantum information is written and read and how it is | **Use case:** Perform a rotation on a qubit<br>**Scenario:** A quantum engineer is asked if a problem can be solved with a quantum computer |

| | | |
|---|---|---|
| | different from digital information. | **Task:** The engineer is able to draw on knowledge of the computing possibilities of qubits to understand if and/or when quantum computers may be applicable |
| **13** | Qubits: a first step towards a visualization of qubits: a circle. Learn how a qubit is written, read, and processed in a geometric picture. | **Use case:** Create a single qubit quantum circuit<br>**Scenario:** A quantum engineer is asked to create a quantum circuit<br>**Task**: Knowing how to work with single-qubit gates and how rotations work, the engineer is able to assemble a quantum circuit |
| **14** | Learn what kets are and how to use them | **Use case:** Communicate quantum states<br>**Scenario:** Two engineers need to be able to mathematically describe a quantum state<br>**Task:** The engineer is able to use their knowledge of kets to write down the state of a qubit using universally accepted notation |
| **15** | Interact directly with the Bloch sphere to represent the full set of qubit states. | **Use case:** Create a single qubit quantum circuit<br>**Scenario:** Create a quantum circuit using the full complement of single qubit quantum gates<br>**Task:** By understanding quantum circuits and how all the single qubit gates operate, the engineer is able to assemble a single qubit quantum circuit |
| **16** | Measurement: The concept of quantum measurement using the Bloch sphere - a visual representation of a qubit of information - is introduced. Explore measurement through interacting with a qubit. | **Use case:** Extract useful results from a quantum system<br>**Scenario:** An engineer is tasked with creating a quantum computer that is able to output a useful result for a given problem<br>**Task:** The engineer is able to make use of the measure operation to extract useful information from a quantum system |

| | | |
|---|---|---|
| **17** | Measurement: Why are quantum measurements random and what rules govern them? Understand how the symbolic notation of kets translate into real measurement outcomes and discover the Born rule. | **Use case:** Maximise the utility of the measurement operation<br>**Scenario:** An engineer wants to maximise the usefulness and fidelity of the measurement operations in their system<br>**Task:** By understanding how noise and the different states of a qubit can affect a measurement, the engineer is able to make best use of the measurement operations in a circuit |
| **18** | Measurement in real quantum systems relevant to quantum computing: trapped ions, atoms, superconducting circuits, and more. | **Use case:** Perform a measurement operation on a quantum computer using a specific architecture<br>**Scenario:** An engineer is asked to perform a measurement on a photonic quantum computer<br>**Task:** By understanding the specifics of how measurement is taken on a photonic quantum computer, the engineer is able to better interpret the results of executing a circuit on that architecture |
| **19** | Measurement: What is the measurement problem and is it really a problem? A brief detour in metaphysics at one end of the spectrum and cold hard calculation at the other. | **Use case:** Understand the differences between classical and quantum measurement<br>**Scenario:** A software developer must calculate the effect of measurement operations on an algorithm for a quantum computer<br>**Task:** The developer draws on their physical understanding and technical intuition of quantum measurement to correctly account for the non-classical effects of measuring quantum systems |
| **20** | Circuits: A quantum circuit is a useful tool to visualize and | **Use case:** Communicate the state of a single-qubit circuit in circle notation |

| | | |
|---|---|---|
| | create quantum instructions or algorithms. We will begin with circuits for a single qubit. | **Scenario:** An engineer must communicate or understand the state of a quantum circuit<br>**Task:** By understanding circle notation, the engineer is able to understand the state of a quantum circuit before, during and after execution |
| **21** | Circuits: Two-qubit gates open a whole new world of quantum possibilities, and are necessary to harness the power of quantum computation. | **Use case:** Communicate the state of a multi-qubit<br>circuit in circle notation<br>**Scenario:** An engineer must communicate or understand the state of a multi-qubit quantum circuit<br>**Task:** By understanding circle notation, the engineer is able to understand the state of a multi-qubit quantum circuit before, during and after execution |
| **22** | Circuits: A real quantum algorithm will involve many qubits. In this topic, we will dial the qubit numbers up... to three at least. | **Use case:** Create circuits on quantum hardware<br>where gate choice is restricted<br>**Scenario:** Create a circuit when some gates are unavailable due to limited topology or other reasons Task: By understanding how unavailable multi-qubit gates are able to be synthesized using other available gates, the engineer is able to perform useful computation on a quantum system with gate restrictions |
| **23** | Circuits: A complete quantum toolbox consists of classical gates and quantum gates. Get familiar with the quantum toolbox and know | **Use case:** Get a range of more useful results from a<br>quantum computer<br>**Scenario:** Use common quantum techniques to assemble and use useful quantum circuits |

| | | |
|---|---|---|
| | which is best suited for the job at hand. | **Task:** An engineer is able to use techniques such as amplitude amplification, which are essential to obtaining useful results from today's quantum computers |
| | Circuits: Strap in we're going full quantum. Use a quantum algorithm to search a database! | **Use case:** Perform a quantum search<br>**Scenario:** An engineer is asked to search a large database for a specific pattern using quantum computation<br>**Task:** By understanding Grover's algorithm, the engineer is able to apply it to find a specific encoded item from a database |
| **24** | Entanglement: The concept of quantum entanglement is considered the most complicated and mysterious topic in all of science.Get a picture of why it's important and how it's used. | **Use case:** make best of use entanglement in a quantum circuit<br>**Scenario:** A software developer must create a circuit that uses entanglement<br>**Task:** By understanding the abilities and limits of entanglement, the engineer is able to use entanglement in their circuit in a more useful way |
| **25** | Entanglement can be made precise using ket notation and qubits. Here students learn to recognize which qubit states are entangled and why. | **Use case:** Describe entanglement<br>**Scenario:** An engineer must identify or communicate entanglement to other engineers of their design team<br>**Task:** By understanding how to describe entanglement in ket notation, the engineer is able to read and communicate effectively with others |
| **26** | Entanglement: Learn how to create and visualize entanglement with a few simple instructions. | **Use case:** Create entanglement<br>**Scenario:** Create entanglement in a circuit<br>**Task:** The engineer is able to create circuits using multiple types of entangled states - an |

| | | |
|---|---|---|
| | | important fundamental skill in the quantum industry |
| **27** | Entanglement: Generate entanglement in real quantum systems relevant to quantum computing: ions, atoms, superconducting circuits, and more. | **Use case:** Perform a entanglement on a quantum computer using a specific architecture<br>**Scenario:** An engineer is asked to perform entanglement on a trapped ion quantum computer<br>**Task:** By understanding the specifics of how entanglement takes place taken on a trapped ion quantum computer, the engineer is able to better interpret the results of executing a circuit on that<br>architecture |
| **28** | Entanglement: Understand why entanglement is so uniquely quantum mechanical and what hidden variables are. | **Use case:** Make use of entanglement as a powerful resource in quantum computers, despite some aspects not being fully understood<br>**Scenario:** An engineer must make practical use of entanglement in a quantum circuit<br>**Task:** Knowing the philosophical debates underpinning entanglement, the engineer is nevertheless able to make use of entanglement to solve a practical computational<br>problem |
| **29** | Noise: We have noisy quantum computers now, and may have them for some time. Are they useless, or can something interesting be done with them? | **Use case:** Apply today's NISQ quantum computers to useful problems<br>**Scenario:** An engineer is asked about the feasibility of a simple quantum AI circuit given today's hardware<br>**Task:** The engineer is able to use their understanding of the limitations of today's hardware to nevertheless begin |

| | | |
|---|---|---|
| | | exploring incorporating quantum computers into AI training data optimisation |
| **30** | Noise: What does the effect of noise look like on qubits? See what noise does to qubit states with the Bloch sphere. | **Use case:** Noise is affecting the execution of a quantum circuit<br>**Scenario:** A quantum circuit is not performing as well as expected and the engineer must diagnose the problem<br>**Task:** By understanding different forms of noise and how they manifest, the engineer is able to diagnose the issue, and design a circuit that most robust to leading noise sources |
| **31** | Noise: How can noise be mitigated? How can instructions be implemented in the presence of noise? Learn to get it under control. | **Use case:** Adjust quantum operations to combat noise<br>**Scenario:** An engineer must decide on a noise mitigation strategy to improve the performance of their circuit<br>**Task:** Knowing the type of noise, they are able to draw on knowledge of mitigation methods for that noise type to begin to combat the effects of that noise source |
| **32** | Noise: Remove errors altogether with error correction. | **Use case:** Fix simple errors using quantum error correction<br>**Scenario:** Noise is affecting a quantum circuit<br>**Task:** Make use of quantum error correcting codes to remove simple errors in a circuit |
| **33** | Control: What is control? Where is it used and why is it helpful? Some insights from the general theory of control can give us intuition over the quantum problem. | **Use case:** Assess the incorporation of quantum control<br>**Scenario:** Control methods are required in order to attain better results from a quantum computation<br>**Task:** The engineer is able to draw on an understanding of available control methods and |

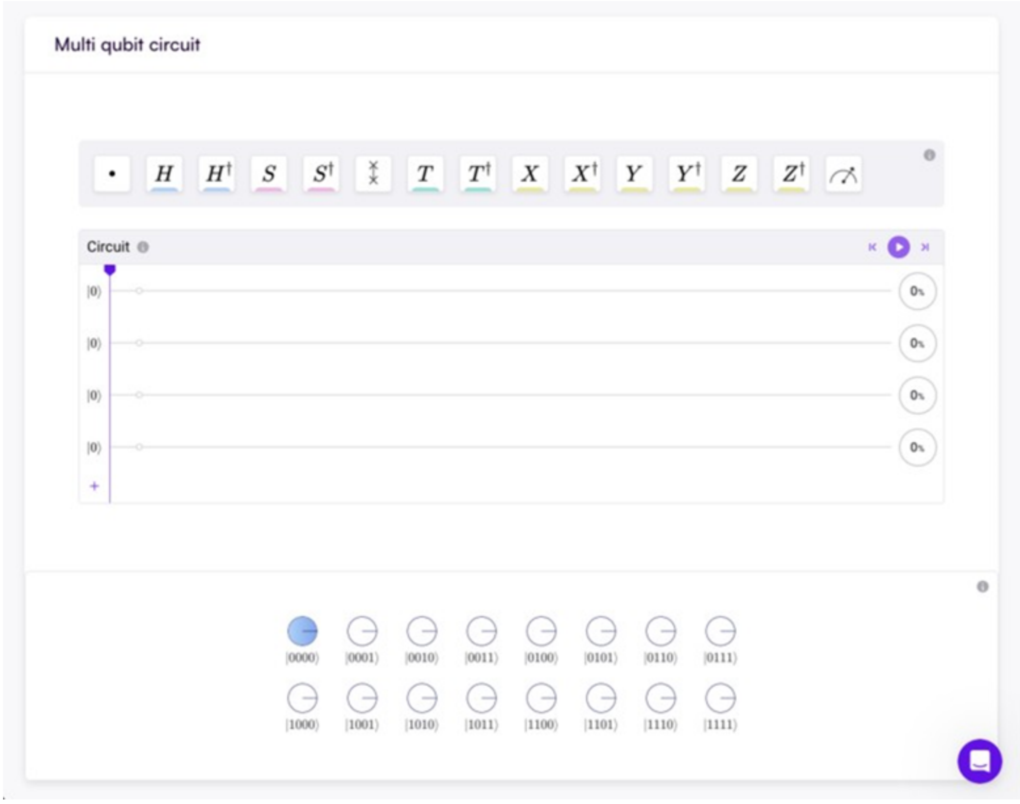
| | | |
|---|---|---|
| | | how they fit into the technology stack to incorporate control into their operations |
| **34** | Control: We know how powerful control can be. In reality, quantum Logic gates are physical operations that take time. Here students will design gates using quantum control. | **Use case:** Incorporate control methods into a quantum circuit<br>**Scenario:** Design a circuit that has increased robustness to noise<br>**Task:** The engineer is able to utilise fundamental quantum control techniques and principles to create a circuit that has increased robustness to noise |
| **35** | Control: Here students will learn about quantum control and how it is used to design quantum gates and algorithms that overcome the fundamental challenge in the field - error. | **Use case:** Incorporate quantum control into an optimisation algorithm<br>**Scenario**: An engineer must implement a quantum optimisation algorithm that is robust to noise<br>**Task:** The engineer is able to draw on their understanding of control to begin to prepare a control system that incorporates AI to improve robustness |
| **36** | Control: How is quantum control used in practice? It is used in every step of the process of solving the problem of creating quantum technology. Here students walk through that process. | **Use case:** Choose a control system<br>**Scenario:** A particular quantum device is subject to fluctuations due to noise<br>**Task:** The engineer is able to choose a control strategy most suitable to combatting that type of noise |
| | Speedup: Proving one algorithm is better than another is not as easy as one might think. Understand some of the key jargon encountered in debates | **Use case:** Describe the complexity of an algorithm<br>**Scenario:** An engineer must communicate the complexity of a given problem or algorithm |

| | | |
|---|---|---|
| | about the utility of quantum computers. | **Task:** Use Big-O notation to easily communicate with others about the class of problem or algorithm being addressed |
| **37** | Speedup: The first quantum algorithm displaying a speedup was dramatic and illustrative. A deep dive into the Deutsch-Jozsa algorithm. | **Use case:** Determine whether an oracle is constant or balanced<br>**Scenario:** Demonstrate that this problem is dramatically faster using a quantum computer<br>**Task:** The engineer is able to implement the the Deutsch-Jozsa circuit to demonstrate quantum speedup |
| | Speedup: Perhaps the most famous algorithm is the factoring algorithm due to Peter Shor, which provides a speedup in the task of factoring large numbers. | **Use case:** Demonstrate the power of Shor's algorithm to break modern cryptography systems<br>**Scenario:** An engineer is asked to verify the impact of quantum computers on common internet cryptography schemes<br>**Task:** The engineer is able to show how Shor's algorithm is demonstrably able to break many cryptographic systems, and will be able to practically do so given a sufficiently powerful quantum computer |
| **38** | Speedup: Non-exponential, yet still significant speedups are known for search and related problems. These might be the first useful applications of quantum computing. | **Use case:** Determine near-term useful applications of quantum computing<br>**Scenario:** An analyst is tasked with evaluating the possible impact of growing quantum computing power on a logistics business<br>**Task:** The engineer is able to show how optimization<br>algorithms are able to solve some problems faster than classical computers, and their capabilities are growing. The engineer can use this |

| | | |
|---|---|---|
| | | knowledge to inform business strategy |
| **39** | Programming: Students are introduced to a set of gates, and must understand how to actually make them operate. | **Use case:** create quantum computing code<br>**Scenario:** A software developer is called upon to write code that implements a particular quantum circuit<br>**Task:** The developer uses knowledge of quantum gates and measurements to correctly generate the code that describes and implements the circuit |
| **40** | Programming: Abstract circuits cannot be directly mapped to real quantum computers. Compiling is the process of translating one quantum program to another, and is how we get quantum computers to calculate. | **Use case:** Assess the properties of quantum compilers<br>**Scenario:** a software developer working in quantum computing needs to choose between available quantum compilers<br>**Task:** The developer uses their knowledge of the quantum compilation process to design metrics against which to assess quantum compilers |
| **41** | Programming: Quantum programming languages allow the rapid development of software to run problem-solving algorithms on quantum computers. Students start with the basics of assembly-level programming. | **Use case:** Code a quantum search algorithm<br>**Scenario:** An engineer is asked to code a quantum search algorithm<br>**Task:** The engineer is able to draw on earlier knowledge of Gover's search algorithm and knowledge of industry-standard QASM code to implement the algorithm |
| **42** | Programming: Quantum computers today are probably running an algorithm with an acronymic name like QAOA. What are | **Use case:** Realize a quantum algorithm on a real quantum computer<br>**Scenario:** A company's high-value optimization problem needs to be solved on a quantum computer |

| | | |
|---|---|---|
| | these and why do they work with noisy quantum computers? Here, students will find out. | **Task:** The student prepares the code that correctly sets up the circuit, in a format that can be used to interact with<br>real quantum computing hardware |

| LIST OF FINAL PROJECTS | |
|---|---|
| **SL. NO** | **FINAL PROJECT** |
| | **T**he Black Opal curriculum includes a final capstone project. The final project draws on knowledge developed throughout the course to tackle a more challenging problem - one that is highly relevant to the emerging quantum computing industry.<br><br>Specifically, the final project instructs students to design a quantum circuit with currently available hardware in mind. Extra care is required to realise the algorithm on a real device architecture, compared to the purely theoretical setting. The student emerges with an understanding of the possibilities (and challenges) of near-term quantum hardware. This knowledge is highly valued by their potential future employers and postgraduate academic programs.<br><br>The project tasks can be explored, tested, simulated, displayed, debugged, and graded all within the Black Opal interactive multi-qubit circuit design and simulation tool (pictured). |

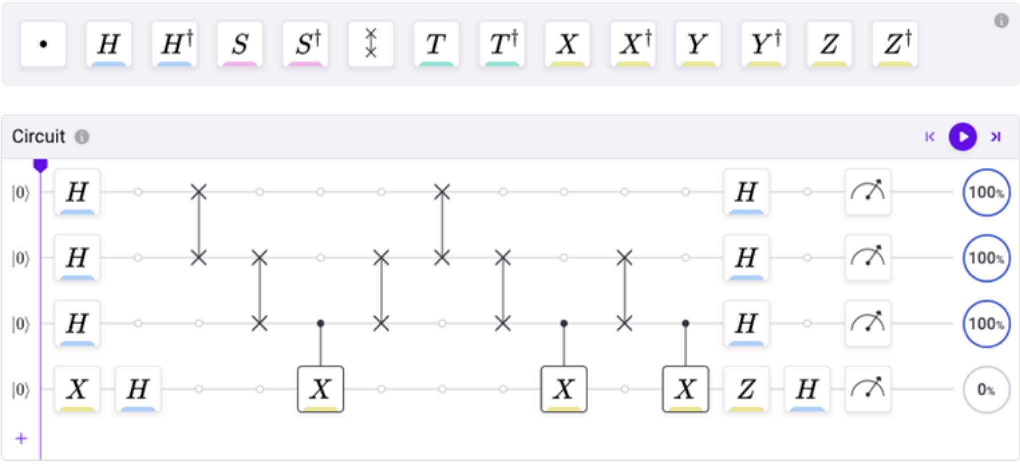



**Task 1**

We consider a four-qubit quantum computer, with qubits labelled #1, #2, #3 and #4. However, due to connectivity limitations on the hardware, we can only implement entangling gates between nearest-neighbours. So CNOT(1,2) and CNOT(2,3) are allowed, but CNOT(1,3) is forbidden, for example. Show how to use multiple gates (four swap gates, and one CNOT gate), to implement an overall effective CNOT(1,4). Hint: Refer to Black Opal lesson 'Programming > Gate, set, match > From A to B' for a refresher on using SWAP gates, and see 'Programming > Compiling > Circuit's end' for a refresher on transpilation with restricted device connectivity.

**Task 2**

Adapt the given quantum circuit to use only allowable gates (Hint: you will have to add SWAP gates. Hint: Refer to Black Opal lesson Programming > Compiling > Decomposed!).




**Task 3**

Layout selection. In the previous circuit, qubit #4 played a special role. It was the target qubit for all the CNOT gates. Was this choice of qubit #4 important? To answer this, redesign the circuit, but swap the role of qubit #4 with another qubit. Next, remove any forbidden gates by adding SWAPs as necessary. Do you get the same number of SWAP gates? What is the fewest total number of SWAP gates? Hint: The answer may depend on which qubit is used as the target for all the CNOT gates!