

Welcome to day four of Quantum in the Summer! My name is Dom, and with me are Sam and Euan, and today we are going to be teaching you all about quantum computing.

For the last few days you've learned all about the bizarre properties of quantum mechanics, and have hopefully even seen some of these for yourselves. These bizarre properties make quantum physics very hard to simulate, to the extent that we build massive supercomputers – such as the one on top of this very building – for the purpose of simulating physical effects. And this irritated a lot of physicists in the 1980s. After all, quantum mechanics describes a collection of physical effects, and a computer is a purely physical device. If simulating these effects on a computer is so hard, why don't we build computers which take advantage of these same effects?

It is from this question that the subject of quantum computing arose: computers which take advantage of quantum physics to solve problems faster. And over the following thirty years, many people have researched what problems quantum computers might be able to solve better than our current machines. From cryptographic attacks to simulation to machine learning, people have proposed many speedups for quantum computers. And alongside this research, groups have been working around the globe to try and build large quantum computers.

But we are jumping the gun here. It's nice seeing the developments of quantum computers, but first we need to understand the basics of what a quantum computer actually is. We shall answer this question by comparing it to a classical computer, such as the machine right in front of you. Fundamentally, a classical computer has two parts to it:

- First, we have data. This data is represented as binary digits, or bits, which can be 0 or 1.
- And secondly, we have operations that we want to apply that data. This is done in the form of logic gates, which take bits as input and produce bits as output.

In quantum computing, we have analogues to these fundamental concepts: We store data in quantum bits, or qubits, and operate on that data using quantum gates. However, there are significant differences between these classical ideas and their quantum versions. For example, quantum bits can, like their classical counterpart, be $|0\rangle$ or $|1\rangle$. but they can also be in a superposition of these two states, where looking at the qubit might say that it is $|0\rangle$, or that it is $|1\rangle$.

Up until 11 we will learn about the fundamentals of quantum computing: what qubits are, examples of some quantum gates, and how we can take advantage of superposition and entanglement to solve problems like generating random numbers or teleporting. After a short break, Sam will guide us through some problems which quantum computers can outperform classical computers in. After lunch, we will finish the workshops off by looking at some ideas behind how you might implement a quantum computer.

To learn about these concepts in an interactive manner, we have developed worksheets for each session for you to work through. For examples, we will be using QCEngine, a simulation of a quantum computer that runs in your own web browser. While it is worth emphasising that you do not need programming experience in order to take part in this workshop, there are some caveats to take note of when programming in general. In particular, try to ensure you copy what is given in the code examples closely, as even changing a lowercase letter to a capital letter can cause your code to not work. Other than that, make your way through the worksheets; ask questions if you are stuck or really curious about something; when writing code try to think of what the code might do before running it; and, most importantly, have fun!