Quave — quantum computing lesson plans

Teacher: ____________ Class: ____________ Dates: ____________

The complete Quave curriculum — all units and lessons, in order. The editable version lives on the Curriculum page at quave.sensym.ai: pick lessons, reword anything, print your own plans. Demos replay exactly from their seeds. Explanations follow Quave's honesty rules and are pending physicist review.

Lesson 1 · Small things play by different rules

Unit 1 · The Quantum World · One 45–60 minute class.

The everyday rules of balls and switches stop working at the scale of atoms and light — and the new rules are about waves that add up or cancel.

STUDENTS WILL BE ABLE TO

- Describe how two waves meeting can make a bigger wave (adding) or a calm spot (cancelling).
- Retell the two-slit story: light makes stripes, and the stripes appear even one photon at a time.
- Explain why scientists needed new rules for very small things.

KEY VOCABULARY

WORD	IN PLAIN WORDS	IN QUBBLE
atom	the tiny building block everything is made of	–
photon	the smallest possible piece of light — light arrives in whole pieces, never half a piece	–
wave	a moving pattern, like ripples on water, that can add up or cancel where two waves meet	–
interference	what happens where waves overlap: crest-on-crest grows, crest-on-trough cancels	–

BACKGROUND FOR THE TEACHER (THE NIGHT-BEFORE READ)

Scale first: atoms are unimaginably small. There are more water molecules in one glass of water than there are glasses of water in all the oceans on Earth. At that scale, our everyday intuition — built on balls, switches, and billiard tables — simply has no experience to draw on.

The clue came from light. In 1801 Thomas Young let light pass through two narrow slits and saw stripes on the far wall — bright, dark, bright, dark. Only waves make stripes: where two ripples meet crest-on-crest they grow, and crest-on-trough they cancel. So light is a wave.

But a century later, detectors showed light arriving in whole pieces — photons — one click at a time, never half a click. Here is the astonishing part: send photons through the two slits one at a time, and the stripes still build up, dot by dot. Each photon lands in one spot, but where it is likely to land follows a wave pattern.

So the honest summary for your class: light is detected in whole pieces, and the chances of where each piece lands follow a wave. A wave of what? Of amplitudes — which is exactly the next lesson. Resist the urge to resolve the tension today; the tension is the physics.

THE HOOK

Two flashlights pointed at the same spot make a brighter spot. But two water waves meeting can make a perfectly calm spot. Which one is light like? (Strangely: both. That puzzle started all of quantum physics.)

MATERIALS

- A shallow pan of water (or a sink) and two fingertips — the whole wave lab.
- A rope or long spring for a second wave demo (optional).
- Board space to draw crest-on-crest and crest-on-trough meetings.

LESSON FLOW

1. Warm-up (8 min): tap the water with one finger — ripples. Now tap with two fingers at once and look between them: some spots dance twice as hard, some spots stay almost calm. Name the two cases: adding and cancelling.
2. Draw it (10 min): on the board, draw two waves meeting crest-on-crest (grows) and crest-on-trough (cancels). Have students predict which meetings grow and which cancel before you draw the result.
3. The stripes story (15 min): tell Young's experiment — light through two slits makes stripes, so light must be a wave. Then the twist: detectors click one whole photon at a time, and the stripes STILL appear, dot by dot, even one photon at a time.
4. Discuss (10 min): "each photon lands in ONE spot, but the pattern of many photons is stripes." What does that tell us about where a photon is likely to land? Let the class sit with the strangeness — do not resolve it.
5. Close (5 min): the rules of the very small are wave rules. Next lesson we learn how physicists WRITE those rules down — with bars you can draw.

HONEST LANGUAGE (WHEN YOU HEAR IT, SAY IT)

You will hear: "So light is just a wave" — or — "light is just a stream of tiny balls."

Say instead: Light is detected in whole pieces (photons), but where those pieces land follows a wave pattern. We need both ideas; neither alone is the full story.

You will hear: "The photon splits in half and goes through both slits."

Say instead: No detector has ever caught half a photon. What passes through both slits is the wave of chances — the pattern that says where the whole photon may land.

PRACTICE

1. `Predict` Two wave crests, each 2 cm tall, meet at one spot. How tall is the water there for that moment? What if a 2 cm crest meets a 2 cm trough?
2. `Read` A friend says: "stripes on the wall prove light is a wave." What would you expect on the wall instead if light were only a stream of tiny balls? (Hint: think of spray paint through two door cracks.)
3. `Discuss` The stripe pattern appears even when photons go through one at a time, over an hour. What is doing the "adding and cancelling" when there is only ever one photon in the room?
4. `Design` Invent your own two-source wave experiment with sound, water, or a spring. Where do you predict the loud/tall spots and the quiet/calm spots?

TEACHING TIPS

- Misconception: students picture waves as stuff sloshing. In the pan, yes — but for light the wave is a pattern of chances. Do not fight this today; plant the question "a wave of WHAT?" and let Lesson 2 answer it.
- Demo move: the two-finger pan tap is the whole lesson in one image. Practice it once beforehand — steady, gentle taps make clearer calm lines than hard splashes.

- Fast finishers: have them measure the spacing between calm lines in the pan and see if it changes when their fingers tap farther apart (it does — closer stripes).

A BIT OF HISTORY

Thomas Young (1801) shined light through two narrow slits and saw stripes — the adding-and-cancelling signature that only waves make. It overturned a century of "light is particles" and, another century later, set up quantum physics' deepest puzzle.

WHY IT MATTERS (REAL WORLD)

- Noise-cancelling headphones make a sound wave that meets the noise crest-on-trough — cancellation you can wear.
- The rainbow shimmer on a soap bubble or a CD is light waves adding and cancelling at different colors.

CHECK FOR UNDERSTANDING

Exit ticket: draw two waves meeting so they cancel. In one sentence, say what you would see at that spot on a screen — and one thing that surprised you about the photon version of the experiment.

Lesson 2 · States and amplitudes: physics on paper

Unit 1 · The Quantum World · One 45–60 minute class.

A "state" is everything you can honestly say about a thing. Quantum states are written as amplitudes — strengths you can draw as bars — and an amplitude's square is that outcome's chance.

STUDENTS WILL BE ABLE TO

- Define a state as "everything you can honestly say about a system right now."
- Draw a quantum state as amplitude bars and read each bar's label and height.
- Use the square rule: an amplitude of 0.71 means a chance of about 0.71 × 0.71 ≈ 1/2.

KEY VOCABULARY

WORD	IN PLAIN WORDS	IN QUBBLE
state	everything you can honestly say about a thing right now	`what plot_state draws`
amplitude	the strength of one possibility in a blend; multiply it by itself to get that outcome's chance	`the height of a bar`
superposition	a blend of possibilities, each with its own amplitude	`what spin creates`
probability	the chance of an outcome, from 0 (never) to 1 (always)	–

BACKGROUND FOR THE TEACHER (THE NIGHT-BEFORE READ)

Start classical: the state of a coin on the table is one fact — heads. The state of a light switch is on or off. Classical states answer every question in advance.

A quantum state is richer: it assigns each possible outcome a strength, called an amplitude. Max Born's rule (1926) says the chance of an outcome is its amplitude multiplied by itself. An amplitude of 1 means certain; 0 means never; 0.71 means about a coin-flip chance, because 0.71 × 0.71 ≈ 0.5.

This is why Quave draws bars. The bars ARE the state written down — one bar per possible outcome, height showing its strength. Nothing is hidden behind the picture; a physicist writing $\alpha|0\rangle + \beta|1\rangle$ on a whiteboard is writing exactly those two bars in symbols.

One honesty guardrail for the whole year: a blend is not "we don't know which it is." A spinning coin is just an unknown; a quantum blend is a real physical state whose parts can later add up or cancel like the waves of Lesson 1. The experiment that tells the two apart is interference — that is Unit 4, and it is worth the wait.

THE HOOK

A weather app says "70% rain." That is a chance. Now imagine you could see WHY: a bar for rain and a bar for sun, and the bars themselves are real physics you can steer. That picture is today's lesson.

MATERIALS

- Board space for drawing bar pairs ($|0\rangle$ and $|1\rangle$) at different heights.

- Optional but recommended: the Quave Studio on the projector for the last 5 minutes (the demo program below).
- Rulers or grid paper if students draw their own bars.

LESSON FLOW

1. Warm-up (5 min): "What is the state of this light switch? Of a flipped-but-covered coin? Of tomorrow's weather?" Sort answers into definite facts vs chances.
2. Introduce amplitude bars (15 min): draw two bars labeled |0⟩ and |1⟩. Full bar on |0⟩, empty on |1⟩: "definitely 0." Then two equal bars: "an equal blend." The funny brackets |⟩ are just physics' way of naming a possibility.
3. The square rule (10 min): amplitude 0.71 on each bar → chance 0.71 × 0.71 ≈ 0.5 each. Let students compute: amplitude 0.87 → about 3/4; amplitude 0.5 → 1/4. Strength times itself is chance.
4. Projector glimpse (5–10 min): run the demo — one line makes a qubit, one line draws its state. The class sees a real amplitude-bar readout: a full bar on |0⟩. This is the exact picture they drew by hand.

   ```
   # Teacher demo: a first look at real amplitude bars.
   q = qubit          # a fresh qubit: definitely 0
   plot_state q       # draw its state — one full bar on |0>
   ```

   > **Predict first:** Before running: we made a fresh qubit and changed nothing. How many bars will have any height, and which one?

5. Close (5 min): a state is the honest summary of a system; amplitude bars are how quantum physics writes one down. Next lesson: what happens when we finally ASK the system for an answer.

HONEST LANGUAGE (WHEN YOU HEAR IT, SAY IT)

> *You will hear: "So the qubit is both 0 and 1 at the same time."*
>
> **Say instead:** It is in a blend of possibilities: part of its amplitude points at 0 and part at 1, and you can see exactly how much of each in the bars. When you measure, you always get exactly one answer.

> *You will hear: "The bars are just a cartoon of what's really going on."*
>
> **Say instead:** The bars are the state, drawn honestly — the same information a physicist writes as α|0⟩ + β|1⟩. Quave never draws a prettier picture than the math.

PRACTICE

1. `Read` A state has a bar of height 1 on |1⟩ and nothing on |0⟩. Describe this state in one plain sentence.
2. `Predict` A state has amplitude 0.87 on |0⟩ and 0.5 on |1⟩. Which outcome is more likely, and roughly what is each chance? (Multiply each strength by itself.)
3. `Write` Draw the amplitude bars for: (a) definitely 0, (b) definitely 1, (c) an equal blend. Label each bar with its amplitude.
4. `Discuss` What is the difference between "the coin is spinning and we don't know yet" and "the qubit is in a blend"? (Hint from Lesson 1: which one could ever show adding-and-cancelling?)

CHALLENGE

The square rule in reverse: a game gives you a 1-in-4 chance of winning. If a qubit's |1⟩ bar meant "win," how tall would its amplitude have to be? (What number times itself is 0.25?)

PROJECTOR DEMO

First amplitude bars — rehearse it once the night before; the pinned seed replays identically in class.

```
# Teacher demo: a first look at real amplitude bars.
q = qubit          # a fresh qubit: definitely 0
plot_state q       # draw its state — one full bar on |0>
```

TEACHING TIPS

- Misconception: "70% rain" feels the same as a quantum blend. The difference is that weather-chance comes from ignorance; a blend is a physical state that can interfere. Name the difference once, promise the proof in Unit 4, and move on.
- Projector move: after the demo, ask "what one thing could we type to make the OTHER bar full instead?" — you do not need the answer today; you are planting Lesson 5's turn and flip.
- Fast finishers: give them the amplitude pairs (0.6, 0.8) and ask them to check that the two chances add up to 1. (0.36 + 0.64 — the bars always account for everything.)

A BIT OF HISTORY

Max Born (1926) realized that an amplitude's square gives the chance of that outcome — a rule so central it earned a Nobel Prize and still governs every quantum computer ever built.

WHY IT MATTERS (REAL WORLD)

- Chemists compute amplitude patterns of electrons to predict how molecules bond — the bars of a molecule are its chemistry.
- Every quantum computer readout in every lab is reported exactly like Quave's bars: outcomes and their measured frequencies.

CHECK FOR UNDERSTANDING

Exit ticket: a qubit has amplitude 0.71 on both |0⟩ and |1⟩. Draw its bars and write each outcome's chance. Bonus sentence: why is this NOT the same as "we just don't know"?

Lesson 3 · Honest randomness: the universe's fair coin

Unit 1 · The Quantum World · One 45–60 minute class.

Quantum randomness is not ignorance in disguise. A coin flip could be predicted with enough cameras; a quantum measurement genuinely has no script — the universe's only honest coin.

STUDENTS WILL BE ABLE TO

- Distinguish practical randomness (a coin flip you failed to track) from honest randomness (no script exists).
- Predict the rough shape of a many-run tally from a state's amplitude bars.
- Explain what the simulator's seed is, and why real quantum machines don't need one.

KEY VOCABULARY

WORD	IN PLAIN WORDS	IN QUBBLE
measurement	asking the system for one definite answer; the blend produces a single outcome	`measure`
outcome	the one answer a measurement gives (for a qubit: 0 or 1)	–
tally	the count of outcomes over many repeated runs	`histogram`
seed	the starting number for the simulator's pretend randomness — same seed, same results, every time	–

BACKGROUND FOR THE TEACHER (THE NIGHT-BEFORE READ)

A coin flip feels random, but it isn't deeply random: with a slow-motion camera and physics, the outcome is decided the moment the coin leaves your thumb. Its randomness is our ignorance.

A quantum measurement is different in kind. When a qubit in an equal blend is measured, the chances (from the amplitude bars) are all there is — a century of looking has found no hidden script underneath. Einstein disliked this ("God does not play dice"); experiment has sided with the dice.

Honesty about our own tool: Quave is a simulator running on an ordinary computer, so its "randomness" is pretend — a formula stirred by a starting number called the seed. Same seed, same run, bit for bit. That is a feature for a classroom (demos reproduce on the projector; everyone can check everyone), and an honest difference from a real quantum machine, which gets its randomness from nature and has no seed at all.

Today's demo measures an equal blend 200 times. Expect roughly half 0s and half 1s — but not exactly half, and the wobble is the point: chance speaks in tallies, not guarantees.

THE HOOK

I can rig any coin flip with a slow-motion camera — the answer is decided before the coin lands. Now: is there ANY coin in the universe that cannot be rigged, even in principle? (Yes. We'll flip it 200 times today.)

MATERIALS

- A real coin for the opening flip.

- The Quave Studio on the projector (demo below).
- Board space for a two-column 0/1 tally.

LESSON FLOW

1. Warm-up (8 min): flip a real coin. Ask: "if I had a super camera and perfect physics, could I predict it?" (Yes.) So coin randomness is ignorance. Write the question of the day: is any randomness NOT ignorance?
2. The honest coin (12 min): run the demo — make a qubit, spin it into an equal blend, measure 200 times. Read the tally together: near half-and-half, never exactly.

```
# The universe's fair coin, flipped 200 times.
q = qubit
spin q                      # equal blend: two bars of 0.71
histogram q, shots: 200     # measure it 200 times and count
```

> **Predict first:** The bars are 0.71 and 0.71 — squares 0.5 and 0.5. Before running: out of 200 flips, how many 1s do you expect? Exactly 100?

3. Rerun and reseed (10 min): run again with the same seed — identical tally (the class should find this suspicious!). Change the seed — a new tally, still near half-half. Name it honestly: the simulator imitates quantum chance with a formula; a real quantum machine needs no seed.
4. Ignorance vs honesty (10 min): back to the board. Coin: script exists, we can't read it. Qubit: bars are the whole story; no script. A century of experiments has looked for a hidden script and found none.
5. Close (5 min): quantum randomness is the only randomness that cannot be rigged — which is why banks and lotteries care. Unit 1 is done: waves, states, honest chance. Next unit, the class starts typing.

HONEST LANGUAGE (WHEN YOU HEAR IT, SAY IT)

You will hear: "The qubit secretly knew its answer all along — measuring just reveals it."

Say instead: Measurement produces a definite outcome from a blend; it does not read off a hidden note written in advance. Experiments have tested the hidden-note idea directly, and nature says no.

You will hear: "The simulator's random numbers are real quantum randomness."

Say instead: Quave imitates quantum chance with a seeded formula so class demos reproduce exactly. A real quantum machine gets its randomness from nature itself — no seed, no formula, no rig.

PRACTICE

1. `Predict` A qubit's bars give chances 3/4 for 0 and 1/4 for 1. Out of 400 measurements, about how many 1s do you expect? Would exactly that number every time make you suspicious?
2. `Read` Two students run the fair-coin demo with the same seed and get identical tallies. One says "the simulator is broken — randomness shouldn't repeat!" What is the honest reply?

```
q = qubit
spin q
histogram q, shots: 200
```

3. `Discuss` Why might a lottery or a bank prefer a random number that even a perfect physicist could not have predicted?
4. `Write` In two sentences: the difference between coin-flip randomness and quantum randomness. Use the word "script" (or "hidden note") at least once.

CHALLENGE

Tally detective: run the fair coin with shots: 10, then shots: 100, then shots: 1000. Which tally lands closest to half-and-half as a fraction? Write the pattern you notice (more shots → steadier fractions).

```
q = qubit
spin q
histogram q, shots: 10
# Try shots: 100 and shots: 1000 too — compare the fractions.
```

PROJECTOR DEMO

The universe’s fair coin — rehearse it once the night before; the pinned seed replays identically in class.

```
# The universe's fair coin, flipped 200 times.
q = qubit
spin q                      # equal blend: two bars of 0.71
histogram q, shots: 200     # measure it 200 times and count
```

TEACHING TIPS

- Misconception: "random" means "anything can happen." Point at the bars: chance has exact strengths; 200 flips of a fair blend will hover near 100/100, never near 200/0. Randomness with rules is the whole subject.
- Projector move: the same-seed rerun is the best gasp of the lesson. Ask the class to call out "rigged!" — then let them explain in what sense it is (simulator) and in what sense the real thing never could be.
- Fast finishers: the challenge (10 vs 100 vs 1000 shots) previews the law of large numbers; have them report fractions, not raw counts.

A BIT OF HISTORY

Albert Einstein and Niels Bohr (1927) argued at the Solvay conference over whether nature truly rolls dice — Einstein insisted a hidden script must exist. A century of ever-sharper experiments has sided with Bohr: no script.

WHY IT MATTERS (REAL WORLD)

- National labs sell certified quantum random numbers for lotteries and security — randomness with a physics guarantee instead of a promise.
- The codes that protect bank cards need numbers nobody could predict; quantum sources are the gold standard for "nobody, even in principle."

CHECK FOR UNDERSTANDING

Exit ticket: give a qubit with chances 1/2 and 1/2, measured 200 times. Write (a) the tally you expect, (b) one sentence on why the coin in your pocket is not honestly random, and (c) what the simulator’s seed does.

Unit 1 rubric — The Quantum World

SKILL	BEGINNING	DEVELOPING	SECURE
Waves that add and cancel	Can say waves sometimes grow and sometimes cancel when they meet	Predicts correctly whether two drawn waves grow or cancel, and retells the two-slit stripes story	Explains why stripes appearing one photon at a time forced physics to new rules
States and amplitudes	Reads which outcome a single full bar means	Draws bars for definite and blended states and applies the square rule to one bar	Checks that squared amplitudes total 1 and explains why a blend is not the same as not-knowing
Honest randomness	Says quantum measurement outcomes are random	Distinguishes coin-flip (ignorance) randomness from quantum randomness, and predicts a rough tally from chances	Explains what the simulator's seed does and why a real quantum machine needs none

Edit any wording to fit your class before printing.

Lesson 4 · Meet the qubit

Unit 2 · The Qubit: a Blend You Can See · One 45–60 minute class.

A bit is a switch: 0 or 1. A qubit is a system whose state is amplitude bars — and a freshly made qubit is honestly, boringly, definitely 0.

STUDENTS WILL BE ABLE TO

- Write and run a first Qubble program: make a qubit and plot its state.
- Read the bar display: labels (|0⟩, |1⟩), heights, and what a full single bar means.
- State the difference between a bit and a qubit in one honest sentence.

KEY VOCABULARY

WORD	IN PLAIN WORDS	IN QUBBLE
bit	a switch that is exactly 0 or exactly 1 — the atom of ordinary computing	–
qubit	a quantum bit: a system whose state is amplitudes over 0 and 1 (say: KEW-bit)	`qubit`
ket |0⟩	physics brackets naming a possibility; the labels under the bars	`the bar labels`
program	a list of instructions the computer follows top to bottom	–

BACKGROUND FOR THE TEACHER (THE NIGHT-BEFORE READ)

Everything an ordinary computer does is bits — billions of switches, each definitely 0 or definitely 1. Every photo, song, and game reduces to those switches.

A qubit is the quantum version. Its state is the amplitude bars from Unit 1: a bar for |0⟩ and a bar for |1⟩. A bit can only ever be "one full bar"; a qubit may hold any honest blend. That is the entire difference — and the entire subject.

Important honesty: a freshly made qubit is NOT mysterious. It starts definitely 0 — one full bar on |0⟩, nothing on |1⟩. Blends must be made on purpose (next lesson). Starting boring is a feature: every program begins from the same known state.

The word "qubit" is young — Benjamin Schumacher coined it in the 1990s. Your students are learning vocabulary that did not exist when their grandparents were in school.

THE HOOK

Every game you have ever played is switches — billions of tiny 0-or-1 switches flipping. Today you get a new kind of switch, one that can hold a blend. First step: meet it while it is still boring.

MATERIALS

- Studio access for every student (or pairs): quave.sensym.ai → Open the Studio.
- The demo program below, typed live — resist pasting; two lines are worth typing.
- Board sketch of the expected bar display for prediction.

LESSON FLOW

1. Orient (8 min): open the Studio on the projector. Point out the three parts: the editor (where code goes), Run, and the bar display. Rule of the room: predict before you run — every time.
2. First program (12 min): type the two lines together. Before running, the class predicts the bars (Unit 1 skill: fresh qubit = definitely 0). Run. One full bar on |0⟩. The picture they drew by hand last week is now live.

```
# My first Qubble program.
q = qubit          # make a qubit named q — it starts definitely 0
plot_state q       # draw its amplitude bars
```

> **Predict first:** A fresh qubit, nothing done to it. Which bar has height, and how tall?

3. Say hello (8 min): add a say line and run again — programs can talk. Then rename q to something else everywhere and confirm nothing changes but the name.

```
q = qubit
say "hello, my qubit is ready"
plot_state q
```

4. Bit vs qubit sentence (10 min): each student writes their one-sentence version of the difference. Collect a few on the board; polish toward: "a bit is 0 or 1; a qubit's state is amplitude bars, which may hold a blend."
5. Close (5 min): today's qubit was definitely 0 the whole time — no blend, no mystery, and that is honest. Next lesson, one new word makes the blend.

HONEST LANGUAGE (WHEN YOU HEAR IT, SAY IT)

> *You will hear: "A qubit is a magic bit that is every number."*
>
> **Say instead:** A qubit is a real physical system whose state is amplitude bars. Today's qubit was definitely 0 the entire lesson — blends exist, but you have to make them on purpose, and you can always see them in the bars.

PRACTICE

1. `Read` What does this program draw? Describe the bars before you run it.

```
z = qubit
plot_state z
```

2. `Predict` If you make TWO fresh qubits and plot each, what does each display show? Try it.

```
a = qubit
b = qubit
plot_state a
plot_state b
```

3. `Fix it` This program should make a qubit and draw it, but it does not run. Find and fix the mistake.

```
q = qubit
plot_state
```

```
q
```

4. `Write` Write a program that makes a qubit named after your favorite letter, says one greeting, and plots the state.
5. `Discuss` Why is it useful that every qubit starts definitely 0, instead of starting random?

CHALLENGE

The say command can print several things separated by commas. Make a program that says your name and then plots two different qubits. What do you notice about their two displays? (Same picture — same honest starting state.)

```
a = qubit
b = qubit
say "made by", "me"
plot_state a
plot_state b
```

PROJECTOR DEMO

Meet the qubit — rehearse it once the night before; the pinned seed replays identically in class.

```
# My first Qubble program.
q = qubit          # make a qubit named q — it starts definitely 0
plot_state q       # draw its amplitude bars
```

TEACHING TIPS

- Misconception: kids expect the first qubit to already be strange. Lean into the boring: "definitely 0, and the bars prove it" builds the trust you will spend in later lessons.
- Projector move: predict-before-run starts today and never stops. Make it a chant if you like — hands up with a bar-shape guess before every Run for the rest of the year.
- Fast finishers: let them break things on purpose — misspell qubit, plot a name that doesn't exist — and read the error messages aloud. Friendly errors are part of the language.

A BIT OF HISTORY

Benjamin Schumacher (1995) coined the word "qubit" for the quantum bit (crediting a conversation with William Wootters) — proof that the vocabulary of this field is younger than many of its students' parents.

WHY IT MATTERS (REAL WORLD)

- Real qubits are made from single atoms, tiny superconducting circuits, or single photons — different hardware, identical amplitude-bar rules.
- Lab machines also initialize every qubit to definitely 0 before a program runs — Quave's boring start is the industry's boring start.

CHECK FOR UNDERSTANDING

Exit ticket: write from memory a two-line program that makes a qubit and draws its state, then sketch the bars it produces. One sentence: bit vs qubit.

Lesson 5 · Spin: an equal blend

Unit 2 · The Qubit: a Blend You Can See · One 45–60 minute class.

One word — spin — turns a definite 0 into an equal blend: two bars of 0.71, each squaring to a half-chance. The blend is real, visible, and honestly described.

STUDENTS WILL BE ABLE TO

- Use spin to create an equal blend and read the resulting bars (0.71 and 0.71).
- Apply the square rule to the blend: each outcome has chance about 1/2.
- Describe the blend honestly: a blend of possibilities with visible amplitudes — measuring gives one answer.

KEY VOCABULARY

WORD	IN PLAIN WORDS	IN QUBBLE
spin (Hadamard)	the move that turns definite 0 into the evenest blend; physicists call it the Hadamard gate	`spin`
equal blend	a superposition with the same strength on 0 and on 1 — amplitude 0.71 each	–
gate	one move applied to a qubit; programs are gates in a row	–

BACKGROUND FOR THE TEACHER (THE NIGHT-BEFORE READ)

The gate called spin (physicists say Hadamard, after Jacques Hadamard's matrices) takes definite 0 to the evenest possible blend: amplitude 0.71 on |0⟩ and 0.71 on |1⟩. Why 0.71? Because chances must add to 1, and $0.71^2 + 0.71^2 \approx 0.5 + 0.5 = 1$. The bars always account for everything.

This is the lesson where the do-not-say rule earns its keep. The tempting shortcut — that the qubit is now two things at once — is both wrong and unnecessary: your class can SEE the true state. Two bars, 0.71 each. Say what the picture says: a blend of possibilities, and measuring will give exactly one answer.

A preview worth planting but not explaining: spin twice and the qubit returns to definitely 0. Blends can un-blend — because amplitudes add and cancel like Lesson 1's waves. The full why is Unit 4 (interference); today it is a magic trick with a promised explanation.

Notation bridge for your strongest students: the state after spin is written $(|0\rangle + |1\rangle)/\sqrt{2}$, and $\sqrt{2} \approx 1.41$, so each amplitude is $1/1.41 \approx 0.71$. The mysterious 0.71 is just "1 divided by the square root of 2."

THE HOOK

Yesterday's qubit was the most boring object in the school: definitely 0. Today, one four-letter word gives it a genuine blend — and you will watch it happen in the bars.

MATERIALS

- Studio access for every student or pair.
- Board sketch: before-bars (full |0⟩) and after-bars (0.71 / 0.71) for the prediction ritual.

- The number 0.71 written large somewhere — it will come up all year.

LESSON FLOW

1. Predict and spin (12 min): start from yesterday's program, add one line. Class predicts the bars first (most will guess two equal bars; heights are the surprise — why not 0.5 each?). Run: 0.71 and 0.71.

```
# One word makes a blend.
q = qubit
spin q              # definite 0 -> equal blend
plot_state q        # two bars: 0.71 and 0.71
```

> **Predict first:** After spin, both outcomes should be equally likely. What height will each bar have — and why might it NOT be 0.5?

2. Why 0.71 (10 min): square rule on the board — chances must total 1, so each amplitude must be the number whose square is 1/2. Let students discover $0.5^2 + 0.5^2 = 0.5 \neq 1$: bars of 0.5 would lose half the universe's probability!
3. Say it honestly (8 min): the class states what the display shows, banning the shortcut phrase before anyone reaches for it: not two things at once — a blend of possibilities with amplitude 0.71 each, one answer upon measuring.
4. The double-spin trick (10 min): predict, then run. Gasp, then promise: the explanation is Unit 4, and it is the reason quantum computers work at all.

```
# The trick: spin twice.
q = qubit
spin q              # blend...
spin q              # ...and un-blend?!
plot_state q        # back to one full bar on |0>
```

> **Predict first:** If spin blends the qubit, what should a second spin do? (Trick question — trust the bars, not your intuition.)

5. Close (5 min): one word makes a blend; the bars never lie about it; and something about blends can cancel. All three of today's facts become superpowers later.

HONEST LANGUAGE (WHEN YOU HEAR IT, SAY IT)

You will hear: "Now the qubit is 0 AND 1 at once!"

Say instead: Look at the display: a blend of possibilities — amplitude 0.71 toward 0 and 0.71 toward 1. It is one state, honestly drawn. When we measure, we will get exactly one answer.

You will hear: "Spin is like shaking the qubit up so it forgets."

Say instead: Spin is a precise, reversible move — so precise that a second spin undoes it perfectly. Forgetting cannot be undone; spin can. The bars just showed us.

PRACTICE

1. `Predict` What bars does this draw? Heights, not just labels.

```
w = qubit
spin w
plot_state w
```

2. `Read` A program plots bars of 0.71 and 0.71. A student says "each measurement will now give half a 0 and half a 1." What is the honest correction?
3. `Fix it` This program wants an equal blend, but it will not run. Fix it.

```
q = qubit
spin
plot_state q
```

4. `Write` Write a program that makes TWO qubits, spins only one of them, and plots both — a definite state and a blend side by side.
5. `Predict` What does spin, spin, spin (three times) leave? Use the double-spin trick to reason it out, then check.

```
q = qubit
spin q
spin q
spin q
plot_state q
```

CHALLENGE

Blend-and-count: combine this lesson with Lesson 3 — spin a qubit and take a 200-shot tally. Does the tally match the squares of the bars? Then try the double-spin version: what does its tally look like, and why is it boring again?

```
q = qubit
spin q
histogram q, shots: 200
# Now try adding a second spin before the histogram.
```

PROJECTOR DEMO

Spin: an equal blend — rehearse it once the night before; the pinned seed replays identically in class.

```
# One word makes a blend.
q = qubit
spin q             # definite 0 -> equal blend
plot_state q       # two bars: 0.71 and 0.71
```

TEACHING TIPS

- Misconception: this is THE lesson where "both at once" tries to enter the room. Do not argue in the abstract — point at the bars every single time. The display is your co-teacher.
- Projector move: run the double-spin with theatrical reluctance ("this can't work..."). The return to |0⟩ lands harder when you play the skeptic.
- Fast finishers: hand them the 0.71 mystery formally — find the number whose square is 1/3, then 1/4. (They are rediscovering square roots; the 1/3 case previews unequal blends.)

A BIT OF HISTORY

Jacques Hadamard (1893) studied the matrices that a century later named quantum computing's most-used move: the Hadamard gate — Qubble's spin. He never saw a qubit; his math was waiting for them.

WHY IT MATTERS (REAL WORLD)

- Nearly every real quantum program starts with a wall of Hadamard gates — spin on every qubit — to open all amplitudes for steering.
- Quantum random-number services are essentially today's demo run on real hardware: spin, measure, repeat.

CHECK FOR UNDERSTANDING

Exit ticket: sketch the bars before and after spin on a fresh qubit, with heights labeled. One sentence using the word "blend" — and not the phrase we banned.

Lesson 6 · Turn: tilt the blend

Unit 2 · The Qubit: a Blend You Can See · One 45–60 minute class.

spin gives one special blend; turn gives them all. Choose an angle, tilt the state, and the square rule tells you exactly the chances you built.

STUDENTS WILL BE ABLE TO

- Use turn with a chosen angle to build unequal blends.
- Connect angle → bars → chances: read amplitudes and square them into a prediction.
- Verify a prediction against a many-shot tally.

KEY VOCABULARY

WORD	IN PLAIN WORDS	IN QUBBLE
turn (rotation)	tilt the state by an angle you choose, in degrees; physicists call it a rotation gate	`turn q, by: 60`
degrees	the angle units on a protractor: 360 all the way around; 180 flips the qubit fully	–
unequal blend	a superposition where one outcome is stronger than the other	–

BACKGROUND FOR THE TEACHER (THE NIGHT-BEFORE READ)

turn q, by: ANGLE tilts the state like a dimmer switch, where spin was only an on/off preset. A small angle leaves the qubit mostly 0; turn by 180 carries it all the way to definitely 1; and 90 degrees lands exactly on the equal blend.

The numbers, for your own preparation (students discover them from the bars): after turn by 60 the amplitudes are 0.87 on $|0\rangle$ and 0.5 on $|1\rangle$ — so the chances are 0.75 and 0.25. In general the $|1\rangle$-chance grows smoothly from 0 (at 0°) to 1 (at 180°). You never need the formula in class; the bars and a tally teach the curve by experiment.

This lesson is where prediction becomes quantitative. The full loop — choose an angle, read the bars, square into chances, tally 400 shots, compare — is real experimental physics: model, prediction, measurement, comparison. Grade the loop, not the arithmetic.

Rabi connection: rotating a real qubit by "just the angle you choose" is done with precisely timed pulses — radio waves for atoms, microwaves for superconducting chips. Isidor Rabi showed how in 1937; MRI machines run on the same physics with the protons in your body.

THE HOOK

spin built one blend — the 50/50. Can we build a 75/25? A 99/1? Today you get a dial instead of a switch: pick an angle, tilt the state, and check your own prediction 400 times.

MATERIALS

- Studio access for every student or pair.
- A protractor image on the board (0°, 60°, 90°, 180° marked).
- A shared table to collect angle → tally results from the class.

LESSON FLOW

1. The dial (12 min): run turn by 60 and read the bars: 0.87 and 0.5. Square them: 75% and 25%. The class just built a rigged-but-honest coin — rigged in the open, where everyone can see the rig in the bars.

   ```
   # A coin you can tilt.
   q = qubit
   turn q, by: 60     # a partial tilt: mostly 0, some 1
   plot_state q       # bars: 0.87 and 0.5 -> chances 3/4 and 1/4
   ```

   > **Predict first:** Sixty degrees is a third of the way to a full flip (180). Will the |1⟩ bar be taller or shorter than 0.71? Will the chances be equal?

2. Check by tally (10 min): add a 400-shot histogram. Compare the tally to the squared prediction: near 300/100. The wobble is Lesson 3's honest randomness, back on schedule.

   ```
   q = qubit
   turn q, by: 60
   histogram q, shots: 400   # predict first: about 300 zeros, 100 ones
   ```

   > **Predict first:** From chances 3/4 and 1/4: out of 400 shots, how many 1s? Write the number down before running.

3. Angle hunt (15 min): pairs pick their own angles (30, 90, 120, 150, 180...) and fill the class table: angle, bar heights, squared prediction, 400-shot tally. The 90° pair discovers spin's twin; the 180° pair discovers the full flip.
4. Read the curve (8 min): plot the class table on the board — |1⟩-chance vs angle. A smooth climb from 0 to 1. One dial, every possible coin.
5. Close (5 min): spin is just one point on today's dial (90°). Next lesson: the pictures physicists use for all this — and which picture to trust.

HONEST LANGUAGE (WHEN YOU HEAR IT, SAY IT)

> *You will hear: "turn by 60 means the qubit is 60% a 1."*
>
> **Say instead:** The angle is the tilt, not the chance. Read the chance from the bars: after turn by 60 the |1⟩ amplitude is 0.5, so the chance is 0.5 × 0.5 = 25%. Angle turns into amplitude; amplitude squares into chance.

PRACTICE

1. `Predict` Before running: which bar is taller after turn by 30, and is the |1⟩-chance more or less than 25%?

   ```
   q = qubit
   turn q, by: 30
   plot_state q
   ```

2. `Read` A classmate's bars show 0.5 on |0⟩ and 0.87 on |1⟩. What chances did they build, and what angle (roughly) did they use? (Hint: compare to 60.)
3. `Fix it` This program wants a tilted coin but will not run. Fix it.

```
q = qubit
turn q, 60
plot_state q
```

4. `Write` Build a coin that lands 1 about one time in four, and prove it with a 400-shot tally. (You already know this angle.)
5. `Design` A board game needs a spinner with a 9-in-10 chance of "advance." Which angle would you try first, and how would you check yourself? (Experiment beats formula — hunt for it.)

CHALLENGE

Two quarter-turns: does turn by 90 twice equal turn by 180 once? Predict from the dial idea, then test both programs. Then the deeper one: does turn by 90 equal spin? Compare their bars — and their 200-shot tallies.

```
a = qubit
turn a, by: 90
turn a, by: 90
plot_state a
# Compare with: b = qubit, turn b, by: 180, plot_state b
```

PROJECTOR DEMO

Turn: a coin you can tilt — rehearse it once the night before; the pinned seed replays identically in class.

```
# A coin you can tilt.
q = qubit
turn q, by: 60     # a partial tilt: mostly 0, some 1
plot_state q       # bars: 0.87 and 0.5 -> chances 3/4 and 1/4
histogram q, shots: 400
```

TEACHING TIPS

- Misconception: angle = percentage. The Fix-it and the honest-language pair both target it; the cure is always the same two-step spoken aloud: "angle → bars, bars → squares."
- Projector move: run the class angle-hunt table live and plot it rough on the board — a hand-drawn smooth curve from class data is more convincing than any printed chart.
- Fast finishers: challenge them to land within ±2 of a 100/300 split by choosing shots and angle — they will discover 120° (the mirror of 60°) on their own.

A BIT OF HISTORY

Isidor Rabi (1937) showed how to rotate an atom's quantum state by exactly the angle you choose using timed radio pulses — the real-world turn. It won the 1944 Nobel Prize and is the working heart of MRI machines.

WHY IT MATTERS (REAL WORLD)

- Every operation on a real quantum chip is ultimately a set of precisely timed turns — quantum programs compile down to pulse schedules.
- An MRI scan is millions of protons in your body being turned by chosen angles and read back — hospital-grade turn q, by: 60.

CHECK FOR UNDERSTANDING

Exit ticket: after turn by 60, write the bar heights, the two chances, and your expected 400-shot tally. One sentence: why is the angle not the same thing as the chance?

Lesson 7 · Pictures of a qubit: the bars tell the truth

Unit 2 · The Qubit: a Blend You Can See · One 45–60 minute class.

Physics uses many pictures for one qubit — bars, spheres, arrows. Helpful pictures are welcome; the amplitude bars are the honest one, and they keep working when the going gets strange.

STUDENTS WILL BE ABLE TO

- Use flip to send a qubit all the way to definitely 1, and read the result in the bars.
- Trace a state through a multi-step program by predicting the bars at each plot_state.
- Explain why Quave treats the amplitude bars — not any prettier picture — as the truth.

KEY VOCABULARY

WORD	IN PLAIN WORDS	IN QUBBLE
flip (NOT gate)	the full half-turn: definite 0 becomes definite 1 and back; the quantum cousin of NOT	`flip`
Bloch sphere	a globe-shaped map physicists use for one qubit's state — one helpful picture among several	–
model	a picture or story that helps thinking; good scientists know each picture's limits	–

BACKGROUND FOR THE TEACHER (THE NIGHT-BEFORE READ)

This lesson closes Unit 2 by teaching a science skill bigger than qubits: telling a helpful picture from the honest record. The amplitude bars are Quave's honest record — they contain everything the math says, no more, no less.

The most famous helper is the Bloch sphere: Felix Bloch's globe where definite 0 sits at the north pole, definite 1 at the south, and blends in between. It makes turn beautifully literal — the angle is a real angle on the globe. Show it, enjoy it, and then say its limit out loud: the globe works for ONE qubit and has no honest way to draw two entangled ones (Unit 5). The bars scale; the globe does not.

flip completes the gate family: the full 180° carry from 0 to 1. Students who did the Lesson 6 angle-hunt have already met it as turn by 180; naming it flip connects to the NOT gate of ordinary computing.

The through-line to leave ringing: "trust the bars." When later units bring stranger states, the bars will keep telling the truth after every helper picture has given up.

THE HOOK

A map of Earth is useful — and wrong (Greenland is not that big). Physics pictures are the same: useful, and wrong somewhere. Today: the pretty globe physicists love, the flip gate, and which picture never lies to you.

MATERIALS

- Studio access for every student or pair.

- A drawing or model of a globe/sphere for the Bloch picture (a real globe works).
- A world map on the wall makes the "useful but distorted" opener land instantly.

LESSON FLOW

1. flip (10 min): predict, then run. One full bar — on |1⟩ this time. Connect to Lesson 6: flip is exactly turn by 180, now with a name and a job (the quantum NOT).

```
# The full flip.
q = qubit
flip q              # definite 0 -> definite 1
plot_state q        # one full bar on |1>
```

> **Predict first:** flip is the whole 180. Where does all the amplitude end up?

2. Trace a journey (12 min): a three-stop program with a plot_state at every stop. Students predict all three displays BEFORE running — this is reading code with physics eyes, the unit's graduation skill.

```
# Three stops, three pictures.
q = qubit
plot_state q        # stop 1: fresh — full bar on |0>
spin q
plot_state q        # stop 2: equal blend — 0.71 and 0.71
flip q
plot_state q        # stop 3: flipped blend — still 0.71 and 0.71!
```

> **Predict first:** Stops 1 and 2 you know. Stop 3 is the puzzle: what does flip do to an EQUAL blend? Can the bars even show a change?

3. The globe (12 min): introduce the Bloch sphere with a real globe: poles are 0 and 1, turn walks you along a meridian, flip is pole-to-pole. Let students place spin's blend (the equator). Then the honest limit, stated by you and repeated by them: one qubit only — it cannot draw two linked qubits. The bars can.
4. Picture court (8 min): quick debate — for each picture (bars, globe, "coin spinning in the air"), students give one way it helps and one way it misleads. (The spinning coin is the misleading one: it smuggles in "secretly decided already," which Lesson 3 debunked.)
5. Close (5 min): Unit 2 graduation — your class can make, blend, tilt, flip, and READ a qubit, and they know which picture to trust. Unit 3 turns to measuring seriously: tallies, seeds, and honest counting.

HONEST LANGUAGE (WHEN YOU HEAR IT, SAY IT)

You will hear: "The globe/arrow IS the qubit."

Say instead: The globe is one helpful map for a single qubit — and like every map, it distorts and it has edges. The amplitude bars are the honest record, and they keep working when qubits link up and the globe cannot.

You will hear: "Stop 3 shows flip did nothing to the blend."

Say instead: The bar heights match, but flip swapped which amplitude belongs to which label. Bars show strengths and phases — and in Unit 4 the phase colors will expose differences the heights alone cannot.

PRACTICE

1. `Predict` What bars does flip-then-spin draw? Compare with spin-then-flip from the flow. Order matters in quantum programs — does it here?

```
q = qubit
flip q
spin q
plot_state q
```

2. `Read` A display shows one full bar on |1⟩. List two different one-gate programs that could have produced it. (One uses flip; one uses turn.)
3. `Fix it` This program should flip a qubit to definite 1, but it will not run. Fix it.

```
q = qubit
flip r
plot_state q
```

4. `Write` Write a program that visits definite 0, definite 1, and an equal blend — in that order — with a plot_state proving each stop.
5. `Discuss` Your friend loves the globe picture. Without banning it, what one warning label would you attach to it?

CHALLENGE

The round trip: using only gates from this unit, take a qubit from definite 0 to definite 1 and back to definite 0 — proving each leg with plot_state. Find two genuinely different routes. (Routes exist through flip, through turn angles, and through spin combinations.)

```
q = qubit
flip q
plot_state q       # leg 1: at |1>
flip q
plot_state q       # leg 2: home again
# Now find a different route — no flip allowed this time.
```

PROJECTOR DEMO

Flip and the honest picture — rehearse it once the night before; the pinned seed replays identically in class.

```
# The full flip.
q = qubit
flip q             # definite 0 -> definite 1
plot_state q       # one full bar on |1>
```

TEACHING TIPS

- Misconception: pictures are either "true" or "false." The mature stance — every picture helps somewhere and misleads somewhere — is the actual lesson; the globe is just today's example.

- Projector move: in the trace activity, cover the display with a sheet of paper until all three predictions are on the board. The reveal-per-stop rhythm keeps every student committed to a guess.
- Fast finishers: stop 3's "flipped blend" looks identical by height. Ask them to hunt for what DID change (watch the phase coloring) — a perfect scout mission for Unit 4.

A BIT OF HISTORY

Felix Bloch (1946) mapped single-qubit states onto a globe — the Bloch sphere — while founding the physics behind MRI. A beautiful, Nobel-winning map… for exactly one qubit, which is why Quave's bars stay the main picture.

WHY IT MATTERS (REAL WORLD)

- Quantum engineers debug real chips with tomography — reconstructing exactly Quave's bars from many measurements, because the bars are the state.
- Choosing the right picture for the job (and knowing its limits) is the daily craft of working scientists — from climate models to subway maps.

CHECK FOR UNDERSTANDING

Exit ticket: sketch the bars after (a) flip on a fresh qubit and (b) spin on a fresh qubit. One sentence: why does Quave call the bars — not the globe — the honest picture?

Unit 2 rubric — The Qubit: a Blend You Can See

SKILL	BEGINNING	DEVELOPING	SECURE
Reads and writes Qubble	Runs a given two-line program and points at the resulting bars	Writes programs using qubit, spin, turn, flip, and plot_state without a template	Fixes broken programs by reading error messages and explains each fix
Predicts before running	Makes a guess about the bars before running when prompted	Predicts bar heights for one-gate programs and checks predictions against tallies	Traces a multi-gate program stop by stop, predicting every plot_state correctly
Says it honestly	Avoids the banned shortcut phrases when reminded	Describes a blend as possibilities-with-amplitudes and measurement as producing one answer	Corrects a misconception aloud (picture limits, angle-vs-chance, hidden-script) with an honest alternative

Edit any wording to fit your class before printing.

Lesson 8 · Measure: one answer from the blend

Unit 3 · Measurement & Probability · One 45–60 minute class.

Measurement is the moment quantum meets everyday: the blend produces exactly one definite outcome, and afterwards the qubit really is that answer.

STUDENTS WILL BE ABLE TO

- Use measure to get a definite 0 or 1 from a blend, and store it with a name.
- Describe honestly what measurement does: produces one outcome from a blend, with chances set by the bars.
- Show with plot_state that after measurement the qubit is definite — the blend is spent.

KEY VOCABULARY

WORD	IN PLAIN WORDS	IN QUBBLE
measure	ask the qubit for one definite answer; the blend produces a single outcome	`result = measure q`
outcome	the one answer you get: 0 or 1	–
collapse	physicists' word for the blend becoming the one definite outcome you observed	–
variable	a name that stores a value so you can use it later	`result`

BACKGROUND FOR THE TEACHER (THE NIGHT-BEFORE READ)

Until now the class has looked at states without touching them: plot_state is a window, not a question. measure is the question. Ask a blended qubit "0 or 1?" and you get exactly one answer, with chances given by the squared bars.

The part that deserves care: after the measurement, the qubit IS the answer. Measure a fair blend, get 1, and a plot_state now shows one full bar on |1⟩. Physicists call this collapse. Run the demo several times: the answer varies run to run, and the after-bars always match the answer.

Say it precisely, because Lesson 3 already armed the class: measurement does not read off a secretly pre-written answer, and it is not "just looking." It is an event — the one place in quantum physics where chance actually happens.

Programming detail worth two minutes: result = measure q stores the outcome in a variable, and say can print several things separated by commas. From here on, programs can act on their answers.

THE HOOK

For two units you have been window-shopping: looking at blends through plot_state without touching. Today you finally get to ASK the qubit a question. Warning: asking changes it — permanently.

MATERIALS

- Studio access for every student or pair.

- Board space for a two-column table: "answer we got" vs "bars afterwards".
- The Lesson 3 exit tickets, if you kept them — today cashes their promise.

LESSON FLOW

1. The question (12 min): build the quantum coin flip. Run it several times (change the seed between runs): the answer varies. One question, one definite answer, chances from the bars.

```
# A quantum coin flip.
q = qubit
spin q                    # fair blend: 0.71 and 0.71
result = measure q        # the question — one definite answer
say "the coin says", result
```

> **Predict first:** The bars are 0.71 / 0.71. Can you predict THIS run’s answer? What CAN you predict?

2. The aftermath (12 min): add a plot_state after the measurement and run a few seeds. The blend is gone; one full bar remains, always matching the printed answer. Fill the board table run by run.

```
q = qubit
spin q
say "answer:", measure q
plot_state q              # after the question: definite, and it matches
```

> **Predict first:** Before running: what will the bars look like AFTER the measurement — still 0.71/0.71?

3. Say it right (10 min): three sentences on the board; the class grades each as honest or not. "Measuring shows what the qubit secretly was" (no — Lesson 3). "Measuring randomly picks from the blend’s chances and makes it definite" (yes). "Measuring destroys the qubit" (no — it is still there, now definite).
4. Measure twice (8 min): students add a SECOND measure of the same qubit and predict it. Same answer, every time — because after the first question the qubit is definite. Chance happens once.
5. Close (5 min): one question, one answer, and the state keeps the receipt. Next lesson: how scientists turn one-at-a-time answers into trustworthy counts.

HONEST LANGUAGE (WHEN YOU HEAR IT, SAY IT)

> *You will hear: "Measuring just reveals what the qubit was all along."*
>
> **Say instead:** Before the question there were two bars — a genuine blend, not a hidden answer. The measurement is where the outcome comes into being; afterwards, and only afterwards, the qubit is definitely that answer.

> *You will hear: "If I measure very gently, I can peek without changing it."*
>
> **Say instead:** There is no gentle enough. Any question that actually gets an answer turns the blend into that answer — it is a law of the physics, not a clumsy tool.

PRACTICE

1. `Predict` This program measures the same qubit twice. What relationship will the two printed answers always have, and why?

```
q = qubit
spin q
say measure q
say measure q
```

2. `Read` A program prints "the coin says 0" and then plots one full bar on |0⟩. A classmate says the plot proves the qubit was 0 all along. Use the two-bars-before picture to answer them.
3. `Fix it` This program wants to store an answer and print it, but it will not run. Fix it.

```
q = qubit
spin q
result = measure
say result
```

4. `Write` Write a program that makes an UNfair coin (use turn), measures it, and prints the answer with a message of your own.
5. `Discuss` Why does "chance happens exactly once, at the measurement" make quantum randomness more trustworthy than a coin you could film in slow motion?

CHALLENGE

Freeze tag: measure a fair blend, then apply spin AFTER the measurement and plot. Is the qubit re-blended? (Yes — gates still work on a definite qubit.) Now predict: measure it again — is the second answer locked to the first?

```
q = qubit
spin q
say "first:", measure q
spin q                      # re-blend the now-definite qubit
plot_state q
say "second:", measure q   # locked to the first, or fresh chance?
```

PROJECTOR DEMO

The quantum coin flip — rehearse it once the night before; the pinned seed replays identically in class.

```
# A quantum coin flip, plus the receipt.
q = qubit
spin q                      # fair blend
result = measure q          # one definite answer
say "the coin says", result
plot_state q                # afterwards: definite, matching
histogram q, shots: 100    # fresh runs: chance again, near 50/50
```

TEACHING TIPS

- Misconception: collapse as destruction. The measure-twice beat and the freeze-tag challenge both show the qubit alive and well — just definite. Keep the word "spent" for the blend, not the qubit.
- Projector move: keep a running board table of (answer, after-bars) across five runs. The 100% match rate is the quiet star of the lesson.
- Fast finishers: ask them why histogram’s tally shows chance even though a measured qubit repeats its answer. (Each shot is a fresh run from the top — the histogram runs the whole program again.)

A BIT OF HISTORY

Werner Heisenberg (1927) showed that measurement in quantum physics is never a passive peek — asking one question sharply always costs you another. The idea that observing participates in the outcome starts here.

WHY IT MATTERS (REAL WORLD)

- Real quantum computers end every program exactly this way: a measurement per qubit, then the classical answer is what gets reported back.
- Quantum key security (Unit 7) works BECAUSE any eavesdropper’s peek is a measurement and leaves fingerprints — no gentle peek exists.

CHECK FOR UNDERSTANDING

Exit ticket: a fair blend is measured and gives 1. Write (a) what plot_state shows right after, (b) what a second measure gives, and (c) one honest sentence about what the measurement did — without the word "reveal".

Lesson 9 · Count like a scientist: shots and tallies

Unit 3 · Measurement & Probability · One 45–60 minute class.

One answer tells you almost nothing; a tally of many answers draws the blend's true shape. Fractions steady as shots grow — that is how scientists trust chance.

STUDENTS WILL BE ABLE TO

- Use histogram with a chosen number of shots and read the resulting tally.
- Convert tallies to fractions and compare them to the squared bars.
- Explain why more shots give steadier fractions, and why a perfect 50/50 tally would be suspicious.

KEY VOCABULARY

WORD	IN PLAIN WORDS	IN QUBBLE
shots	how many times the whole program is run and measured for one tally	`shots: 400`
tally	the outcome counts across all shots	`what histogram draws`
fraction	count divided by total shots — the number to compare with the chances	–
sample	the collection of runs you actually took, standing in for the endless runs you could take	–

BACKGROUND FOR THE TEACHER (THE NIGHT-BEFORE READ)

Lesson 8 got one answer per question. Science needs the shape of the chances, and shape only shows in repetition. histogram runs the whole program from the top, N times, and counts.

The key discipline is fractions, not counts. 93 ones out of 400 shots is the fraction 0.23 — and THAT is what should sit near the squared bar (0.25 for a 120-degree turn's |0⟩ side… or is it the |1⟩ side? Let the class fight it out with the bars in front of them).

The second discipline is expectation of wobble. Chance never owes you 300/100 exactly; it owes you "near, and nearer as shots grow." A class that internalizes "wobble shrinks like clockwork but never to zero" has learned more statistics than many adults.

Today cements the experiment loop from Unit 2 — predict from bars, square into chances, tally, compare — as the standard move of the course. Every later unit (interference, entanglement, the case studies) reuses it verbatim.

THE HOOK

I flipped a coin once and got heads. Is the coin fair? (You cannot know.) I flipped it 400 times and got 391 heads. NOW do you know something? Counting is how chance confesses.

MATERIALS

- Studio access for every student or pair.

- Calculators or mental-math confidence for count → fraction conversions.
- A class chart: shots on one axis, distance-from-expected-fraction on the other.

LESSON FLOW

1. Set the trap (10 min): a 120-degree coin. Predict from the bars first: which side is the 3/4 side? Run the tally and settle it with fractions.

```
# A 120-degree coin — which side is loaded?
q = qubit
turn q, by: 120            # past the fair point of 90
plot_state q               # read the bars: 0.5 on |0>, 0.87 on |1>
histogram q, shots: 400    # about 100 zeros, 300 ones
```

> **Predict first:** At 60 degrees, |0⟩ had the 3/4 chance. At 120 — past the fair point — who gets 3/4 now?

2. Fractions on the board (10 min): collect five pairs' tallies. Convert each to fractions. All different counts, all fractions hugging 0.25/0.75 — the chart makes the point no lecture can.
3. The shots ladder (15 min): pairs run the same coin at shots: 10, 100, 1000 and chart the wobble. Small samples lie confidently; big samples settle. Have each pair report their worst small-sample lie ("we got 8 ones out of 10!").
4. Suspicion training (8 min): show a fake tally of exactly 300/100 three runs in a row. Real chance wobbles — a perfect repeat with different seeds should smell wrong. (Same seed, of course, repeats exactly — Lesson 10 makes that a superpower.)
5. Close (5 min): one answer is an anecdote; a tally is evidence; a fraction is the honest summary. This loop — predict, square, tally, compare — is now the course's standard move.

HONEST LANGUAGE (WHEN YOU HEAR IT, SAY IT)

You will hear: "400 shots at 25% means we will get exactly 100 ones."

Say instead: Chance owes you NEAR 100, not 100 — and it gets proportionally nearer as shots grow. A tally that hits the bullseye every single time would actually be evidence something is rigged.

PRACTICE

1. `Predict` From this program's bars, what fraction of 500 shots should be 1s? Give a range you would accept as fair.

```
q = qubit
turn q, by: 60
histogram q, shots: 500
```

2. `Read` Pair A reports 27 ones in 100 shots; pair B reports 241 in 1000. Which result is stronger evidence the chance is 1/4, and why?
3. `Fix it` This program wants a 200-shot tally but will not run. Fix it.

```
q = qubit
spin q
```

```
histogram q, shots 200
```

4. `Write` Build a coin whose 1-fraction should land near 0.1 (hint: hunt with small angles), and prove it with a 1000-shot tally.
5. `Design` Design a fairness test for a mystery coin program you are not allowed to read: what shots would you use, what fractions would you accept, and what would make you cry foul?

CHALLENGE

The wobble census: run the fair coin at shots: 100 with five different seeds, and record the five 1-fractions. Then do the same at shots: 1000. Compare the spreads — write one sentence about what a bigger sample buys you.

```
q = qubit
spin q
histogram q, shots: 100
# Rerun with seeds of your choice, then switch to shots: 1000.
```

PROJECTOR DEMO

The 120-degree coin — rehearse it once the night before; the pinned seed replays identically in class.

```
# A 120-degree coin — which side is loaded?
q = qubit
turn q, by: 120            # past the fair point of 90
plot_state q               # bars: 0.5 on |0>, 0.87 on |1>
histogram q, shots: 400    # about 100 zeros, 300 ones
```

TEACHING TIPS

- Misconception: counts vs fractions. Students will compare raw counts across different shot totals; the board chart in fractions is the cure. Insist tallies are always reported as fractions.
- Projector move: the fake perfect tally is worth staging seriously — let them catch it. Skepticism about too-perfect data is a life skill wearing a physics costume.
- Fast finishers: the 0.1-coin hunt (practice Write) has no listed angle on purpose; they will bracket their way to ~37 degrees by experiment, which is the entire scientific method in miniature.

A BIT OF HISTORY

Jacob Bernoulli (1713) proved the law of large numbers — the theorem that fractions settle toward the true chance as trials grow. Published after his death, it is the exact reason your tallies steady.

WHY IT MATTERS (REAL WORLD)

- Real quantum computers report every result as a shots-and-tally readout — a lab paper’s figure and Quave’s histogram are the same picture.
- Polls, drug trials, and game balance testing all live on Bernoulli’s law: enough samples, honest fractions, expected wobble.

CHECK FOR UNDERSTANDING

Exit ticket: a blend has chances 0.25/0.75. Give (a) your expected tally out of 400 shots, (b) a wobble range you would accept, and (c) one sentence on which you trust more — 10 shots or 1000 — and why.

Lesson 10 · Seeds: replay the randomness

Unit 3 · Measurement & Probability · One 45–60 minute class.

The simulator's chance comes from a formula stirred by a seed — so any run can be replayed exactly. Pretend randomness is a confession AND a classroom superpower.

STUDENTS WILL BE ABLE TO

- Demonstrate that the same program with the same seed reproduces results exactly, and a different seed gives a fresh run.
- Use seeds for honest comparisons: change one thing, keep the randomness fixed.
- State clearly where simulator randomness and real quantum randomness differ.

KEY VOCABULARY

WORD	IN PLAIN WORDS	IN QUBBLE
seed	the starting number that steers the simulator's pretend randomness	`the seed box in the Studio`
pseudo-random	made by a formula that only LOOKS random — with the seed, fully predictable	–
reproducible	a result anyone can re-create exactly — the backbone of trustworthy science	–
fair comparison	changing one thing while holding everything else — including the randomness — fixed	–

BACKGROUND FOR THE TEACHER (THE NIGHT-BEFORE READ)

Lesson 3 confessed it; today makes it a tool. Quave's randomness is a formula (a tiny one, called Mulberry32) stirred by the seed. Same seed, same program: identical answers, identical tallies, bit for bit, on any computer.

This is a confession — a real quantum machine has no seed; nature's chance cannot be replayed — and simultaneously the best experimental tool in the course. Rehearse a demo at home with seed 7, and seed 7 replays it identically on the projector. A student who claims "my program never gives 11" can hand you their seed, and you can watch their exact run.

The deeper science habit is the fair comparison. Want to know what adding a phase does? Run with and without it ON THE SAME SEED: any difference is your change, not luck. Professionals call this controlling variables; kids call it fair.

Field note for honesty: real labs also crave reproducibility — they publish methods, calibrations, and full tally data precisely because their randomness cannot be replayed. Seeds are the simulator's version of open science.

THE HOOK

Yesterday I flipped the universe's honest coin 200 times. Today I will flip it 200 times again — and I promise you, in writing, the exact same 200 answers. How can both lessons be true?

MATERIALS

- Studio access for every student or pair.
- A visible place to write the class's "seed of the day".
- Two-column board space for a with-change vs without-change fair comparison.

LESSON FLOW

1. The promise (10 min): write a seed on the board, predict "identical", run the fair coin twice on it. Identical tallies. Change the seed: fresh run, same half-half character. Name the mechanism: formula + seed.

```
# The replayable coin.
q = qubit
spin q
histogram q, shots: 200    # same seed -> this exact tally, forever
```

> **Predict first:** Run 1 and Run 2 use the same seed. Predict run 2's tally to the digit.

2. Superpower inventory (8 min): the class lists what replay buys: rehearsed demos, checkable homework, debuggable "impossible" results, shareable exact experiments. (All four are real Quave workflows.)
3. Fair comparison lab (15 min): does phase change a lone blend's tally? Same seed, two programs differing by one line. Identical tallies — phase alone changed nothing measurable. Foreshadow: "so is phase useless? Unit 4 says otherwise."

```
# Fair test: does phase alone change the counts?
q = qubit
spin q
phase q                    # the one changed line — delete me for run B
histogram q, shots: 200
```

> **Predict first:** Same seed, one extra phase line. Will the tally shift?

4. Honest borders (8 min): two-sentence summary per student: what seeds CAN do (replay the simulator) and what they can never do (replay a real quantum machine). Trade and grade for honesty.
5. Close (5 min): Unit 3 complete — the class can question a qubit, count like scientists, and replay any experiment. Unit 4 finally spends the phase mystery.

HONEST LANGUAGE (WHEN YOU HEAR IT, SAY IT)

You will hear: "Seeds prove the randomness was fake, so the physics is fake too."

Say instead: The seed is honest about the simulator: its chance is a formula. The physics it imitates — bars, squares, tallies — matches what real quantum machines do; only the SOURCE of the chance differs. Real machines draw theirs from nature, seedless.

PRACTICE

1. `Predict` Two classmates on opposite sides of the room run this with the same seed. How will their tallies compare?

   ```
   q = qubit
   turn q, by: 60
   histogram q, shots: 300
   ```

2. `Read` A student reports: "with seed 4 my coin gave 14 ones out of 20 — the simulator is biased!" Design the two follow-up runs that would settle this fairly.
3. `Write` Build a fair-comparison pair: one program, then the same program with exactly one added turn, both run on one chosen seed. Report what changed and what stayed.
4. `Discuss` A lab's quantum computer results can never be replayed run-for-run. What do real scientists publish instead so strangers can trust them?
5. `Design` Invent a "seed handshake" for the class: a ritual for how two students share a claim (program + seed + expected tally) so the receiver can verify it in under a minute.

CHALLENGE

Seed detective: pick a secret seed, run the fair coin at 20 shots, and give a partner only the program and your tally. Can they FIND your seed by hunting? What does their answer (and how long it took) teach about formulas, seeds, and secrecy?

```
q = qubit
spin q
histogram q, shots: 20
# Partner: hunt the seed that reproduces the tally exactly.
```

PROJECTOR DEMO

The replayable coin — rehearse it once the night before; the pinned seed replays identically in class.

```
# The replayable coin.
q = qubit
spin q
histogram q, shots: 200    # same seed -> this exact tally, forever
```

TEACHING TIPS

- Misconception: "fake randomness" discredits the whole simulator. The honest frame — the CHANCES are physics, the SOURCE is a formula — needs saying once, clearly, then demonstrating often.
- Projector move: the written-in-advance tally prediction (from your rehearsal at home, same seed) is the most convincing sixty seconds in the unit. Let a student verify digit by digit.
- Fast finishers: the seed-detective challenge scales — 20 shots is findable, 200 makes brute force miserable. That gradient IS an intro to why secrets built on big search spaces hold.

A BIT OF HISTORY

John von Neumann (1946) built the first pseudo-random generators for the earliest computers — and joked that anyone treating formula-made digits as true chance lives "in a state of sin." Quave sins openly, with a seed box.

WHY IT MATTERS (REAL WORLD)

- Game worlds, weather models, and machine-learning training runs all pin seeds so results can be reproduced and bugs replayed.
- Scientific journals increasingly require published code AND seeds — reproducibility is now a rule of good science, not a courtesy.

CHECK FOR UNDERSTANDING

Exit ticket: your program with seed 12 prints tally 96/104. Write (a) tomorrow’s tally with seed 12, (b) what you can say about tomorrow’s tally with seed 13, and (c) one sentence on why a real quantum machine could not make promise (a).

Unit 3 rubric — Measurement & Probability

SKILL	BEGINNING	DEVELOPING	SECURE
Questions a qubit honestly	Uses measure and reads the single outcome	Predicts post-measurement bars and explains why a second measure repeats the answer	Explains collapse without "reveal" language and designs a program that shows it
Counts like a scientist	Runs a histogram and reads raw counts	Converts tallies to fractions and compares them to squared bars	Chooses shot counts deliberately and reasons about expected wobble versus suspicious perfection
Uses seeds for honest science	Knows the same seed replays a run exactly	Uses fixed seeds to make fair one-change comparisons	Articulates what seeds can and cannot promise, including the simulator-vs-real-machine difference

Edit any wording to fit your class before printing.

Lesson 11 · Phase: the hidden dial

Unit 4 · Phase & Interference · One 45–60 minute class.

Every amplitude carries a hidden direction — its phase. Flip it and nothing measurable changes… while the qubit is alone. The dial is real, silent, and waiting.

STUDENTS WILL BE ABLE TO

- Use phase and observe: bar heights and tallies are unchanged, but the bar's phase marking flips.
- Extend the wave picture: an amplitude has a size AND a direction, like a crest versus a trough.
- Explain why a lone phase is invisible to measurement (chances come from squares, and squaring hides sign).

KEY VOCABULARY

WORD	IN PLAIN WORDS	IN QUBBLE
phase	the direction of an amplitude — crest or trough of the wave; invisible to a lone measurement	`phase`
sign	plus or minus: two amplitudes of the same size can point opposite ways	–
hidden variable of the honest kind	phase is really there in the state — hidden only from single-qubit tallies, not from physics	–

BACKGROUND FOR THE TEACHER (THE NIGHT-BEFORE READ)

Recall Lesson 1: a wave has crests and troughs, and cancellation happens when crest meets trough. The amplitude bars have carried this direction all along — the display marks each bar's phase — but nothing so far has made it matter.

phase flips the sign of the |1⟩ part: the blend (0.71, 0.71) becomes (0.71, −0.71). Heights identical. Squares identical — because a negative number squares to a positive one. So chances identical, tallies identical (your Lesson 10 fair comparison already proved it, on one seed).

So is phase physically meaningless? No — and the class has seen the proof without naming it: spin-then-spin returns to |0⟩ (Lesson 5's trick), while spin-then-phase-then-spin will land on |1⟩ (next lesson). Two states identical to every lone measurement, sent to opposite destinations by the same gate. Only a real, physical difference can do that.

Frame the unit's arc for yourself: today establishes the dial exists and is silent; tomorrow turns the silence into the loudest effect in quantum computing. Resist merging the two lessons — the cliffhanger does real pedagogical work.

THE HOOK

I am going to change this qubit — really change it — and then defy you to detect the change. Every tally you run will be identical. And on Thursday, this same invisible change will flip the whole answer.

MATERIALS

- Studio access for every student or pair.
- The Lesson 1 crest/trough board drawing, redrawn where everyone can see it.
- Seeds agreed in advance for the fair-comparison reruns.

LESSON FLOW

1. The invisible change (12 min): blend, plot, phase, plot. Heights identical; the phase marking on the |1⟩ bar flips. Name what changed: the direction of the amplitude, crest to trough.

   ```
   # Flip the hidden dial.
   q = qubit
   spin q              # blend: 0.71 and 0.71
   plot_state q        # look at the phase marking on each bar
   phase q             # flip the sign of the 1 part
   plot_state q        # same heights — different direction!
   ```

   > **Predict first:** Will the bar HEIGHTS change? Will anything on the display change?

2. Why silence (10 min): squares on the board: 0.71^2 and $(-0.71)^2$ are both 0.5. Squaring erases sign — so no lone tally can hear the dial. Connect to Lesson 10's fair test: the class already proved this experimentally.
3. Crest and trough (10 min): back to the water pan mentally: two crests meeting grow; crest and trough cancel. Ask: "if amplitudes have directions, what SHOULD be possible when two paths meet?" Let the class predict cancellation — they are one lesson early, on purpose.
4. Two identical strangers (8 min): pose the puzzle formally: plus-blend and minus-blend give identical tallies forever, alone. Are they the same state? Vote, defend, and leave it unresolved on the cliff.
5. Close (5 min): the dial exists, physics keeps the receipt even when tallies cannot, and Thursday the receipt gets cashed.

HONEST LANGUAGE (WHEN YOU HEAR IT, SAY IT)

> *You will hear: "Phase does nothing — we proved it with tallies."*
>
> **Say instead:** Phase does nothing a LONE measurement can hear. The state genuinely changed — the display marks it — and the moment two paths meet, this silent direction decides what adds and what cancels.

PRACTICE

1. `Predict` Two phases in a row: what state is this at the end, and what would any tally show?

   ```
   q = qubit
   spin q
   phase q
   phase q
   plot_state q
   ```

2. `Read` A classmate claims the plus-blend and minus-blend are "the same state with different decorations." Give the strongest one-sentence argument you currently have against this — and note what evidence is still missing.

3. `Fix it` This program wants to flip the hidden dial of a blend, but it will not run. Fix it.

   ```
   q = qubit
   spin q
   phase
   plot_state q
   ```

4. `Write` Write the Lesson-10-style fair comparison yourself: the same seed, with and without one phase line, tallied. Report the (non-)difference honestly.
5. `Discuss` Where else in life does a real difference hide from one kind of test but show up in another? (Two batteries inserted + and – ways still light a bulb alone…)

CHALLENGE

Dial hunt: does phase do anything to a FRESH qubit (no spin first)? Predict from the wave picture — the |1⟩ part of a fresh qubit has zero amplitude, so flipping its direction should flip… nothing. Test with plots and a tally, then explain in one sentence why the dial needs a blend to bite.

```
q = qubit
phase q            # flip the sign of... nothing?
plot_state q
histogram q, shots: 100
```

PROJECTOR DEMO

The hidden dial — rehearse it once the night before; the pinned seed replays identically in class.

```
# Flip the hidden dial.
q = qubit
spin q             # blend: 0.71 and 0.71
plot_state q       # note the phase marking
phase q            # flip the sign of the 1 part
plot_state q       # same heights — different direction!
```

TEACHING TIPS

- Misconception: invisible = nonexistent. The display's phase marking is your ally — the change is literally on screen even while every tally stays silent. Point at it often.
- Projector move: run the two plot_states side by side and have students spot the one difference, quiz-show style. The find is more memorable than the tell.
- Fast finishers: the challenge (phase on a fresh qubit) is a genuine conceptual test — flipping the direction of a zero-height bar. Ask for their explanation in wave words.

A BIT OF HISTORY

Augustin-Jean Fresnel (1818) built the wave theory of light on amplitudes with directions — and predicted, to skeptics' delight-turned-horror, a bright spot in the middle of a shadow. The directions were real; the spot is there.

WHY IT MATTERS (REAL WORLD)

- Noise-cancelling headphones ARE a phase flip: same sound, opposite direction, silence where they meet.
- Telescope arrays and 5G antennas steer beams purely by adjusting phases — silent dials, loud results.

CHECK FOR UNDERSTANDING

Exit ticket: after spin then phase, write (a) the two amplitudes with signs, (b) what any 400-shot tally shows and why, and (c) your current answer to "same state or different?" with one honest sentence of evidence.

Lesson 12 · Interference: the cancel trick

Unit 4 · Phase & Interference · One 45–60 minute class.

Send a blend's paths back together and the hidden directions decide everything: aligned amplitudes add, opposed ones cancel. Chance evaporates — the answer is certain.

STUDENTS WILL BE ABLE TO

- Run the spin–phase–spin sequence and verify the outcome is certainly 1 (and spin–spin certainly 0).
- Explain the result with the wave picture: paths recombine; same-direction parts add; opposite-direction parts cancel.
- Recognize interference as the resolution of the phase cliffhanger: the silent dial decides the loud outcome.

KEY VOCABULARY

WORD	IN PLAIN WORDS	IN QUBBLE
interference	when paths meet: amplitudes pointing the same way add; opposite ways cancel	`what the second spin does`
constructive	adding up — the surviving answer	–
destructive	cancelling out — the vanished answer	–
certain	probability 1: chance has left the building	–

BACKGROUND FOR THE TEACHER (THE NIGHT-BEFORE READ)

The payoff lesson. spin spreads the state into two paths; the second spin brings the paths back together; and what survives depends entirely on the phases. No phase in between: the $|1\rangle$ contributions oppose and cancel, and the qubit returns to certain 0 (Lesson 5's trick, finally explained). With phase in between: the cancellation swaps sides — certain 1.

Let the class feel the strangeness properly: we took a fair coin, flipped one INVISIBLE switch, and the coin became certain. Not 99% — certain, every seed, every run. Only wave arithmetic does this; no story about ignorance or hidden scripts survives contact with it.

This is also the promised proof from Lesson 2 that a blend is not "we don't know": ignorance cannot cancel. If the qubit were secretly 0-or-1, the second spin would act on that secret value and give a fair tally. It gives certainty instead. The blend's two parts were both physically there — their collision proves it.

Name the big picture plainly: THIS is the engine of quantum computing. Arrange phases so wrong answers arrive opposed (cancel) and right answers aligned (add). Every famous algorithm is a clever way to do exactly that; nothing more magical is hiding underneath.

THE HOOK

Thursday's promise, due today: the invisible change that no tally could hear will now flip a certain 0 into a certain 1. Chance itself is about to get cancelled.

MATERIALS

- Studio access for every student or pair.
- The crest/trough drawing AND the two-slit story from Lesson 1 — today closes that loop explicitly.
- Board space for the two-paths diagram (0-path and 1-path recombining).

LESSON FLOW

1. Cash the cliffhanger (12 min): run the full sequence. One bar standing — on |1⟩ — and a measurement to prove it prints 1. Rerun on three seeds: certain, certain, certain.

   ```
   # The cancel trick.
   q = qubit
   spin q              # split into two paths
   phase q             # flip the 1-path's direction
   spin q              # paths meet: 0s cancel, 1s add
   plot_state q        # one bar standing
   say "always:", measure q
   ```

   > **Predict first:** The dial was flipped between the spins. Where does the amplitude end up — and is there any chance left?

2. The arithmetic (12 min): trace it on the board in four numbers. After spin: (0.71, 0.71). After phase: (0.71, −0.71). The second spin makes |0⟩ get (0.71 – 0.71) → 0, |1⟩ get the aligned sum → 1. Cancellation is subtraction; certainty is the leftovers.
3. The ignorance test (10 min): run the control — spin, spin, no phase: certain 0. Then the argument: if the blend were secret ignorance, the second spin would hit a definite-but-unknown bit and tallies would stay fair. Certainty falsifies the hidden script. Lesson 2's IOU, paid.
4. Two slits, again (8 min): rename the pieces: slits = the two paths, screen = the second spin's output, stripes = the certain outcome. The class has now COMPUTED the experiment they only heard as a story in Lesson 1.
5. Close (3 min): wrong answers can be made to cancel. Hold that sentence — Unit 7 builds real algorithms out of it.

HONEST LANGUAGE (WHEN YOU HEAR IT, SAY IT)

> *You will hear: "The computer checks both paths and picks the better answer."*
>
> **Say instead:** Nothing checks or picks. The amplitudes flow through both paths and ADD like waves — opposed parts cancel on their own. The certainty falls out of arithmetic, not judgment.

> *You will hear: "So quantum computers just cancel wrong answers for any problem."*
>
> **Say instead:** Arranging the cancellation is the hard part — it takes a clever setup that exists only for SOME problems. That is why quantum computers help with certain tasks and do nothing special for others.

PRACTICE

1. `Predict` Phase BEFORE the first spin instead of between: trace the four numbers and predict the ending. Then run it.

```
q = qubit
phase q
spin q
plot_state q
histogram q, shots: 200
```

2. `Read` A friend says the cancel trick "proves the qubit was secretly 1 all along." Use the no-phase control run to dismantle this in two sentences.
3. `Fix it` This program should end certainly at |1⟩, but it will not run. Fix it — then check it really is certain.

```
q = qubit
spin q
phase q
spinq
plot_state q
```

4. `Write` Write the two programs of the lesson (with and without phase) and a two-line lab report: which answer canceled in each, in wave words.
5. `Discuss` Why is "wrong answers cancel" a better slogan for quantum computing than "it computes many answers"? Which half of the work does each slogan hide?

CHALLENGE

The double dial: TWO phases between the spins. Trace the numbers first — two sign flips on the 1-path — then predict and run. One sentence: why does an even number of flips restore the original trick?

```
q = qubit
spin q
phase q
phase q            # flip the flip
spin q
plot_state q
say "always:", measure q
```

PROJECTOR DEMO

The cancel trick — rehearse it once the night before; the pinned seed replays identically in class.

```
# The cancel trick.
q = qubit
spin q             # split into two paths
phase q            # flip the 1-path's direction
spin q             # paths meet: 0s cancel, 1s add
plot_state q       # one bar standing
say "always:", measure q
```

TEACHING TIPS

- Misconception: certainty came from measurement somehow "settling" things. The plot BEFORE the measure already shows one bar — the certainty is in the state; the measurement just reads a foregone conclusion.
- Projector move: the three-seed certainty run ("certain, certain, certain") lands the no-chance-left point better than any single run. Milk it.
- Fast finishers: hand them turn-by-90 instead of the second spin and let them discover it does NOT complete the trick — the recombination must match the split. A perfect prelude to how delicate real algorithms are.

A BIT OF HISTORY

Richard Feynman (1948) rebuilt quantum physics as "sum over paths": every route a particle can take contributes an amplitude, and reality is their interference. Your four-number trace is Feynman's method, pocket-sized.

WHY IT MATTERS (REAL WORLD)

- Atomic clocks and gravitational-wave detectors (LIGO) are interference machines — cancellations so precise they can time light and hear spacetime.
- Every quantum algorithm's speedup — factoring, searching, simulating molecules — is engineered interference: wrong answers opposed, right answers aligned.

CHECK FOR UNDERSTANDING

Exit ticket: trace spin–phase–spin in four amplitudes and circle the cancellation. One sentence: what does the certain outcome prove about whether the blend was "really there"?

Lesson 13 · Steering amplitudes: where the power comes from

Unit 4 · Phase & Interference · One 45–60 minute class.

Quantum programs win exactly when phases can be arranged so wrong answers cancel and right answers add. That is the whole superpower — rarer and more honest than the headlines.

STUDENTS WILL BE ABLE TO

- Contrast the three two-spin machines (nothing, phase, double-phase between) and predict each ending from phase bookkeeping.
- State the honest slogan of quantum computing: engineered interference for SOME problems, not universal speed.
- Judge claims about quantum computers using the add/cancel test.

KEY VOCABULARY

WORD	IN PLAIN WORDS	IN QUBBLE
algorithm	a recipe of steps that solves a problem	`your program`
amplitude steering	choosing gates so that, at the end, amplitudes align on right answers and oppose on wrong ones	–
speedup	when the quantum recipe needs far fewer steps than any known ordinary recipe — true for some problems only	–

BACKGROUND FOR THE TEACHER (THE NIGHT-BEFORE READ)

A consolidation-and-honesty lesson before the toolbox grows. The class now owns the mechanism: split (spin), steer (phase, turn), recombine (spin), and the phases decide the ending. Today they drive it themselves and then zoom out to what it is actually FOR.

The interferometer playground: between two spins, insert nothing / one phase / two phases and predict each ending by counting sign flips on the 1-path (even count → certain 0, odd → certain 1). The class is now programming with a quantity no measurement can see directly — worth saying out loud, slowly.

Then the honesty block, which this course treats as content, not disclaimer. Quantum computers are NOT faster at everything; most tasks (email, games, spreadsheets) get zero help. The wins are problems whose structure lets phases be arranged cleverly: searching (Grover — right answers add a little per round), factoring (Shor — a hidden rhythm made to interfere), simulating molecules (nature speaking its native language). No cancellation trick known → no speedup. Period.

Give your class the claim-testing reflex: when a headline says "quantum computer does X," ask — what cancels? If the article cannot say, the claim is decoration. This single question inoculates against 90% of quantum hype, and professionals use exactly it.

THE HOOK

You now hold the actual engine of a trillion-dollar technology: making wrong answers cancel. Today you drive it — and learn to smell when someone selling "quantum" has never looked under the hood.

MATERIALS

- Studio access for every student or pair.
- Board table for the interferometer results: dials set / prediction / outcome.
- Two or three real (or realistic) quantum headlines for the claim-testing game.

LESSON FLOW

1. The interferometer (15 min): pairs run the three machines — nothing, phase, phase-phase between the spins — predicting each from sign counting before running. Fill the board table. The even/odd rule emerges from their own data.

   ```
   # Interferometer, dial at one flip.
   q = qubit
   spin q              # split
   phase q             # dial: one sign flip on the 1-path
   spin q              # recombine
   plot_state q
   say "lands on:", measure q
   ```

   > **Predict first:** Count the sign flips on the 1-path, then call the ending: certain 0 or certain 1?

2. Say the mechanism (8 min): each pair writes the machine's rule in one sentence using add/cancel words. Board the best. ("The dials set directions; the second spin makes directions collide; collisions decide.")
3. What it is FOR (12 min): the honesty block. Three genuine wins (search, factoring, molecules) each told as "what cancels"; three non-wins (email, games, spreadsheets) told as "nothing to cancel." The slogan goes on the wall: interference makes right answers add up and wrong ones cancel — and only some problems let you set that up.
4. Claim court (8 min): read 2–3 quantum headlines; the class cross-examines each with "what cancels?" and issues verdicts: mechanism, plausible, or decoration.
5. Close (2 min): Unit 4 done — the silent dial, the loud collision, and the honest boundary. Unit 5 doubles the qubits, and the strangeness.

HONEST LANGUAGE (WHEN YOU HEAR IT, SAY IT)

> *You will hear: "Quantum computers will make all computers obsolete."*
>
> **Say instead:** They accelerate the specific problems where interference can be engineered — searching, factoring, simulating nature. For most everyday computing there is nothing to cancel, so there is no quantum win. Both halves of that sentence are the truth.

PRACTICE

1. `Predict` Three phases between the spins. Even or odd? Call the ending before you run.

   ```
   q = qubit
   spin q
   ```

```
phase q
phase q
phase q
spin q
plot_state q
```

2. `Read` A headline: "New quantum chip processes information in many states simultaneously." Apply the what-cancels test and write your verdict in two sentences.
3. `Write` Build an interferometer whose ending is certain 0 but which uses at least two phase lines. Prove certainty with plot and tally.
4. `Design` Design a poster titled "What quantum computers actually do" with exactly three panels: split, steer, collide. Caption each in kid-honest words.
5. `Discuss` Why might a company WANT to say "does everything faster" instead of "cancels wrong answers for some problems"? Who pays for the difference?

CHALLENGE

Mixed dials: what does a turn by 180 between the spins do — is it the same dial as phase? Trace what turn does to each path, predict, and run. (Careful: turn moves amplitude BETWEEN bars; phase only redirects. The endings differ in an instructive way.)

```
q = qubit
spin q
turn q, by: 180    # a mover, not a dial — or is it?
spin q
plot_state q
histogram q, shots: 200
```

PROJECTOR DEMO

The two-spin interferometer — rehearse it once the night before; the pinned seed replays identically in class.

```
# Interferometer, dial at zero flips (control).
q = qubit
spin q             # split
spin q             # recombine, nothing steered
plot_state q       # certain 0 — the even-flips ending
say "lands on:", measure q
```

TEACHING TIPS

- Misconception: the superpower is the blend itself. Keep hammering the corrective: blends alone are just fancy coins — the power is the CANCELLATION, and cancellation must be engineered per problem.
- Projector move: claim court works best with one real vendor headline; adjudicating actual marketing with a physics test is empowering in a way invented examples are not.
- Fast finishers: the mixed-dials challenge rewards careful tracing (turn by 180 sends $|0\rangle \to |1\rangle$ and $|1\rangle \to -|0\rangle$); accept any correct account in wave words — the minus sign is the treasure.

A BIT OF HISTORY

Lov Grover (1996) found the quantum search recipe: each round nudges phases so the right answer’s amplitude grows a little while wrong ones fade. A perfect example of the whole art — no magic, just relentless engineered adding-up.

WHY IT MATTERS (REAL WORLD)

- Chemistry teams at pharma and battery companies are the most eager real users — molecules are interference problems in nature’s own format.
- National agencies are already switching codes because Shor’s interference recipe will eventually threaten today’s encryption — policy moving on the strength of a cancellation argument.

CHECK FOR UNDERSTANDING

Exit ticket: (a) the even/odd flip rule in your own words, (b) one problem quantum computers genuinely help and the one-phrase reason ("what cancels"), and (c) one problem they do not help and why not.

Unit 4 rubric — Phase & Interference

SKILL	BEGINNING	DEVELOPING	SECURE
Tracks the hidden dial	Knows phase flips a sign without changing bar heights	Traces amplitudes with signs through short programs and predicts tallies	Explains why a lone phase is invisible (squares hide sign) yet physically real
Reads interference	Runs the cancel trick and reports the certain outcome	Predicts spin–dial–spin endings with the even/odd flip rule before running	Explains cancellation as wave arithmetic and uses it to refute the hidden-script story
Judges quantum claims	Repeats the honest slogan (right answers add, wrong ones cancel, some problems only)	Applies the "what cancels?" test to a claim with prompting	Cross-examines real headlines independently and sorts mechanism from decoration

Edit any wording to fit your class before printing.

Lesson 14 · Two qubits: four possibilities

Unit 5 · Two Qubits & Entanglement · One 45–60 minute class.

Add a qubit and the bars multiply: two qubits blend over four outcomes at once — 00, 01, 10, 11 — and the display becomes a joint story, not two solo ones.

STUDENTS WILL BE ABLE TO

- Predict and read four-bar displays for two-qubit programs.
- Compute joint chances for independent qubits by multiplying (½ × ½ = ¼ per corner).
- Read two-outcome labels MSB-first (first-made qubit is the left digit) without stumbling.

KEY VOCABULARY

WORD	IN PLAIN WORDS	IN QUBBLE
joint state	the state of the pair together — one set of bars over 00, 01, 10, 11	–
independent	neither qubit's outcome tells you anything about the other's; joint chances just multiply	–
basis label	the two-digit name under each bar; the first-made qubit is the LEFT digit	`00 01 10 11`

BACKGROUND FOR THE TEACHER (THE NIGHT-BEFORE READ)

Growth day. One qubit had two bars; two qubits have four — one per pair of answers. plot_state with no name draws the whole system's joint bars, and reading them is today's skill.

Spin both qubits and all four bars stand equal at height 0.5 (chance ¼ each). Two ideas hide in that number: joint chances of independent events multiply (½ × ½), and amplitude 0.5 squares to ¼ — the square rule survives scaling, and the four squares still total 1.

Drill the label order early: the first-made qubit is the LEFT digit. In the 10 bar, qubit a says 1 and qubit b says 0. Thirty seconds of drilling here saves a semester of off-by-one confusion — this exact convention question is where most textbooks lose people.

Keep today deliberately unspooky: spin a, spin b gives genuinely independent coins — measuring a tells you NOTHING about b. Tomorrow, link breaks exactly this, and the contrast only lands if independence was actually felt today.

THE HOOK

One qubit: two bars. Two qubits: four. Three: eight. Somewhere around fifty qubits the bars outnumber every grain of sand on Earth. Today, the first doubling — and learning to read the new labels.

MATERIALS

- Studio access for every student or pair.
- A 2×2 grid on the board (a-answer × b-answer) to map onto the four bar labels.
- Coins for the classical warm-up (two per pair).

LESSON FLOW

1. Classical warm-up (8 min): pairs flip two real coins 20 times and tally the four outcomes HH/HT/TH/TT. Roughly even quarters. This chart is about to be beaten by two lines of code.
2. The four bars (12 min): spin both, plot the joint state. Four bars at 0.5. Square them: quarters, matching the coin chart. Then the label drill: which bar is "a=1, b=0"? (10 — left digit is a.)

   ```
   # Four possibilities.
   a = qubit
   b = qubit
   spin a              # blend the first
   spin b              # blend the second
   plot_state          # all four bars: 00 01 10 11
   ```

   > **Predict first:** Four outcomes, all fair and independent. How many bars, and how tall is each?

3. Independence check (12 min): measure a, then tally b. Whatever a said, b's tally stays fair — knowing a buys you nothing about b. Name it: independent. (Secret purpose: tomorrow this exact experiment gives the opposite result.)

   ```
   a = qubit
   b = qubit
   spin a
   spin b
   say "a said:", measure a
   histogram b, shots: 200    # b doesn't care what a said
   ```

   > **Predict first:** If a came out 1, does b's tally tilt? What does "independent" predict?

4. Half-blends (8 min): spin only a, leave b fresh. Predict the joint bars first: only 00 and 10 can stand (b is certainly 0). The joint display encodes single-qubit certainty as absent bars.
5. Close (5 min): bars multiply, labels read left-first, and independent qubits mind their own business. Hold onto that last sentence overnight — tomorrow one word ends it.

HONEST LANGUAGE (WHEN YOU HEAR IT, SAY IT)

> *You will hear: "Two spun qubits hold four answers, so the computer is already computing four things."*
>
> **Say instead:** The pair holds amplitudes over four possibilities — a richer state, yes, but one measurement still yields exactly ONE pair of digits. Holding possibilities is not yet computing with them; that takes the interference engineering from last unit.

PRACTICE

1. `Predict` Spin only b (not a). Which joint bars can stand, and how tall? Run to check your reading of the labels.

   ```
   a = qubit
   b = qubit
   ```

```
spin b
plot_state
```

2. `Read` A four-bar display shows 0.71 on 00 and 0.71 on 01, nothing else. Describe each qubit in one sentence. (Careful with the left-digit rule.)
3. `Fix it` This program wants two blended qubits but will not run. Fix it.

```
a = qubit
b = qubit
spin a, b
plot_state
```

4. `Write` Build a pair whose joint tally lands near 25/25/25/25 out of 100 shots, and then a pair whose tally is all in ONE corner. Prove both.
5. `Predict` Flip a, spin b: which corners can appear in a 200-shot joint tally?

```
a = qubit
b = qubit
flip a
spin b
histogram a, b, shots: 200
```

CHALLENGE

The unfair pair: build a two-qubit state whose 11-corner gets about 3/4 of a 400-shot tally while the OTHER three corners share the rest. (Hint: what angle gives one qubit a $\sqrt{3}/2$ amplitude on $|1\rangle$ — and what must the other qubit be doing?) Prove it with the joint histogram.

```
a = qubit
b = qubit
turn a, by: 120    # tune me
flip b             # and rethink me
histogram a, b, shots: 400
```

PROJECTOR DEMO

Four possibilities — rehearse it once the night before; the pinned seed replays identically in class.

```
# Four possibilities.
a = qubit
b = qubit
spin a             # blend the first
spin b             # blend the second
plot_state         # all four bars: 00 01 10 11
```

TEACHING TIPS

- Misconception: reading 10 as "ten" or as b-first. The 2×2 grid mapped onto the labels, done once physically at the board, fixes it. Then quiz mercilessly for two minutes.
- Projector move: run the coin tally and the code tally side by side — the "two lines of code beat our 20 flips" moment buys goodwill for the abstraction.
- Fast finishers: the unfair-pair challenge quietly requires them to realize b must be certain-1 (flip) while a is tilted — composing two ideas without being told to. Let them struggle productively.

A BIT OF HISTORY

John von Neumann (1932) wrote the rulebook still in use for how quantum systems combine — the joint state over all outcome pairs. Your four-bar display is a page of his book, rendered live.

WHY IT MATTERS (REAL WORLD)

- Every quantum chip datasheet counts qubits precisely because each one doubles the joint state — the resource IS the bar count.
- The multiply-chances rule for independent events is the same math behind weather forecasts and insurance — now visible as bar heights.

CHECK FOR UNDERSTANDING

Exit ticket: for spin a, flip b — draw or list the possible joint bars with heights, give each corner’s chance, and translate the 01 label into a plain sentence about a and b.

Lesson 15 · Link: the Bell pair

Unit 5 · Two Qubits & Entanglement · One 45–60 minute class.

One link line ends independence: the pair enters a shared state where only 00 and 11 survive. Neither qubit has its own story anymore — the PAIR is the state.

STUDENTS WILL BE ABLE TO

- Build a Bell pair with spin + link and read its two-bar joint display.
- Verify by tally that the qubits always agree, while each alone remains a fair coin.
- State the no-story property honestly: an entangled qubit has no separate state of its own.

KEY VOCABULARY

WORD	IN PLAIN WORDS	IN QUBBLE
link (CNOT)	a two-qubit move: the second qubit flips exactly when the first one's part says 1; physicists call it controlled-NOT	`link a, b`
entangled	sharing one joint state that cannot be split into separate solo stories	–
Bell pair	the famous simplest entangled state: half 00, half 11 — always agreeing	–
correlated	the outcomes move together — compare notes and they match	–

BACKGROUND FOR THE TEACHER (THE NIGHT-BEFORE READ)

Yesterday's closing sentence dies today. spin a opens two paths; link a, b writes a's path onto b: along the 0-path b stays 0, along the 1-path b flips to 1. Joint bars: 00 and 11 at 0.71 each. The corners 01 and 10 are simply GONE — the pair cannot disagree.

The delicate honest point, and the reason your memory of this unit matters: after link, qubit a alone has NO amplitude bars of its own. Ask "what is a's state?" and the honest answer is "that question has no answer — only the pair has a state." Quave's display honors this: the entangled group plots jointly. This is enforced honesty, not a display limitation.

Yet each side alone behaves as a perfect fair coin: measure only a across many runs — fair tally. Measure only b — fair tally. The agreement lives ONLY in the comparison. Sit with how strange that is: two fair coins, anywhere apart, that always match when the notes come together.

Rerun Lesson 14's independence check verbatim on the Bell pair: yesterday a's answer bought nothing about b; today it buys everything. Same experiment, one added line, opposite result — the cleanest before/after in the entire course.

THE HOOK

Yesterday you proved measuring a tells you nothing about b — independence, certified. Today I add ONE line to yesterday's program and re-run your own experiment. Watch your certificate burn.

MATERIALS

- Studio access for every student or pair.
- Yesterday's independence-check program, kept open for the before/after.
- Two volunteers for the "sealed envelopes vs shared blend" skit (see tips).

LESSON FLOW

1. The one-line coup (12 min): insert link into yesterday's program. Joint plot: two bars only — 00 and 11. The disagreeing corners are not small; they are absent.

   ```
   # The famous Bell pair.
   a = qubit
   b = qubit
   spin a                      # open two paths
   link a, b                   # write a's path onto b
   plot_state                  # only 00 and 11 remain
   histogram a, b, shots: 200 # they always agree
   ```

   > **Predict first:** link makes b copy a's path. Which of the four corners can survive?

2. Always agree, each alone fair (10 min): read the 200-shot tally — roughly half 00, half 11, zero disagreements. Then tally a alone: fair coin. The pattern lives in the pair, not in either member.
3. The no-story rule (10 min): try to plot a by itself and discuss what Quave shows for an entangled member. State the rule and its name-brand consequence: an entangled qubit has no solo state — the pair is the smallest thing that HAS a state. (This rule is enforced by the platform because it is enforced by nature.)
4. Envelope court (10 min): the skeptic's comeback — "someone wrote matching letters in two sealed envelopes; agreement is old news." Concede it fully for TODAY's evidence, then plant the flag: physicists found a test that tells shared-blend from sealed-envelopes, and envelopes lose. Tomorrow's lesson.
5. Close (3 min): one line ended independence. The corners didn't shrink; they vanished. And the best question in the room — "how is that different from envelopes?" — is exactly tomorrow's physics.

HONEST LANGUAGE (WHEN YOU HEAR IT, SAY IT)

> *You will hear: "After link, a and b are both 0 and both 1 until we look."*
>
> **Say instead:** The PAIR is a blend over two joint possibilities — all-0s and all-1s, amplitudes visible in the two bars. Each measurement of the pair yields one definite pair of matching digits. Blend language, applied to the pair, stays honest.

> *You will hear: "link just copies a onto b."*
>
> **Say instead:** Along each path, yes — but a was in a blend of paths, so the RESULT is a shared blend, not a copy of a definite value. True copying of an unknown qubit is actually forbidden by the physics (worth telling your class: nature has an anti-copying law).

PRACTICE

1. `Predict` Flip a AFTER the link. Which two corners survive now, and what does the tally show?

```
a = qubit
b = qubit
spin a
link a, b
flip a
histogram a, b, shots: 200
```

2. `Read` A tally over 400 shots: 00×198, 11×202, others zero. Write the two honest sentences: one about each qubit alone, one about the pair.
3. `Fix it` This program wants a Bell pair but will not run. Fix it.

```
a = qubit
b = qubit
spin a
link a b
plot_state
```

4. `Write` Build the DISAGREEING twins: a pair whose tally shows only 01 and 10. (One well-placed flip does it.) Prove with plot and tally.
5. `Discuss` Each side of a Bell pair alone is a fair coin. Where does the always-agree pattern live, if it is in neither qubit? Try to say it without pointing at anything.

CHALLENGE

The three-way pact: chain links to a third qubit so the joint display shows only 000 and 111 (all three always agree). Then check by tally, and answer: does measuring ONE of the three still give a fair coin alone?

```
a = qubit
b = qubit
c = qubit
spin a
link a, b
link b, c              # extend the pact
plot_state
histogram a, b, c, shots: 200
```

PROJECTOR DEMO

The Bell pair — rehearse it once the night before; the pinned seed replays identically in class.

```
# The famous Bell pair.
a = qubit
b = qubit
spin a                      # open two paths
link a, b                   # write a's path onto b
plot_state                  # only 00 and 11 remain
histogram a, b, shots: 200 # they always agree
```

TEACHING TIPS

- Misconception: entanglement as ongoing communication between the qubits. Keep the wave frame: link merged their paths INTO one shared state at contact — the agreement was written then, in amplitudes, not radioed later. (But do NOT let this slide into "sealed envelopes" — hold the tension until tomorrow.)
- Projector move: the before/after on yesterday's own independence check is the emotional core. Run yesterday's program first from the saved seed, then add the one line, live.
- Fast finishers: the disagreeing-twins write-up has two valid solutions (flip before or after the link, on either qubit — let them find and compare two).

A BIT OF HISTORY

John Stewart Bell (1964) took entanglement out of the philosophy lounge by finding a measurable difference between shared quantum states and any sealed-envelope story. The pair that always agrees carries his name.

WHY IT MATTERS (REAL WORLD)

- Entangled pairs are the working currency of quantum networks now running between cities — link generation and delivery is an actual engineering job title.
- Error correction inside every serious quantum computer is built from many-qubit agree-pacts — your three-way challenge, industrialized.

CHECK FOR UNDERSTANDING

Exit ticket: for the Bell pair, write (a) the joint bars with heights, (b) what a 200-shot joint tally shows, (c) what a-alone's tally shows, and (d) the no-story rule in one sentence.

Lesson 16 · Correlated — but silent

Unit 5 · Two Qubits & Entanglement · One 45–60 minute class.

Bell-pair agreement is instant at any distance, yet carries no signal: each side alone sees only fair coins. Nature lets outcomes match without letting anyone talk.

STUDENTS WILL BE ABLE TO

- Demonstrate that either side of a Bell pair, alone, is an unrigged fair coin — no setting on one side tilts the other side's tally.
- Explain why matching without signaling is possible: the pattern appears only when both sides' notes are compared.
- Distinguish honestly between sealed-envelope agreement and shared-blend agreement, and say which tests told them apart.

KEY VOCABULARY

WORD	IN PLAIN WORDS	IN QUBBLE
signal	anything one person can do that another, far away, can DETECT — the thing entanglement never provides	–
no-signaling rule	the theorem that entanglement correlations cannot carry information from one side to the other	–
local	what one side can see using only their own qubit and tallies	–
Bell test	the experiment that distinguishes shared quantum blends from any pre-written-answers story	–

BACKGROUND FOR THE TEACHER (THE NIGHT-BEFORE READ)

The lesson that saves your students from a lifetime of sci-fi misinformation. Set the stage maximally: split a Bell pair between two labs, any distance apart. Measure both; compare notes later: always matching. The temptation — instant matching means instant messaging — is strong, famous, and wrong.

The killer argument fits in one question: what does the FAR side actually see? A fair coin. Before their partner measures, after their partner measures, whatever their partner does: fair coin, identical tallies, nothing to detect. The match is real, but it only EXISTS in the comparison — and the comparison requires carrying notes together by ordinary means, at ordinary speed. Run the demo: whatever is done to a, b's lone tally never moves.

Then pay Lesson 15's envelope debt honestly. For always-agree alone, envelopes ARE a perfect explanation — concede it. The difference appears when each side can choose among different measurement angles: quantum pairs win a specific matching game more often than ANY pre-written answers could. Bell found the scorecard (1964); Clauser, Aspect, and Zeilinger ran ever-cleaner versions and the pre-written story lost every time (Nobel Prize, 2022). Your class cannot run angled measurements in Qubble — say so plainly, and tell them what the grown-up experiments found.

Close the unit with the honest miracle, which needs no inflation: nature permits perfect coordination without communication. Stranger than the myth, and true.

THE HOOK

Two labs, one Bell pair, a thousand kilometers. The answers always match. So... tap out a message in measurements? Every sci-fi writer says yes. Nature says no — and today you will see exactly where the message dies.

MATERIALS

- Studio access for every student or pair.
- The board split into "LAB A" and "LAB B" halves for the whole lesson.
- A prepared pair of "sealed envelopes" (matching letters inside) as the rival theory's prop.

LESSON FLOW

1. The temptation (8 min): set the two-labs scene; collect signaling schemes from the class ("measure in a pattern!", "measure or don't!"). Write them on the board — today we execute and destroy them.
2. Kill the schemes (15 min): the far side's view. Run b's lone tally with a untouched, a measured, a flipped-then-measured: three identical fair-coin tallies. Nothing lab A chooses is detectable at lab B. Scheme by scheme, cross them off.

```
# What does the far lab actually see?
a = qubit
b = qubit
spin a
link a, b                  # the pair splits: a stays, b travels far
flip a                     # lab A frantically "signals"...
say "lab A sent:", measure a
histogram b, shots: 200    # ...lab B sees: a fair coin. Always.
```

> **Predict first:** Lab A flipped and measured. Will lab B's lone tally tilt even slightly?

3. Where the match lives (8 min): now compare columns — run the joint tally: perfectly correlated (after A's flip: always disagree, just as perfectly). The pattern is in the TOGETHER view only, and bringing tallies together takes a truck, a fiber, a walk down the hall: ordinary speed.
4. Envelopes on trial (10 min): hand over the envelope prop — for everything shown TODAY, envelopes explain it. Then tell the Bell-test story as promised debt: choose-your-angle games, quantum score beats every possible envelope, decades of experiments, 2022 Nobel. Verdict: shared blend, not hidden notes — measured, not assumed.
5. Close (4 min): the unit's three facts in a row — pairs can share one state; the sharing coordinates perfectly; the coordination carries nothing. Say the last one in do-not-say form: correlated, but it cannot carry a message.

HONEST LANGUAGE (WHEN YOU HEAR IT, SAY IT)

You will hear: "Entangled particles talk to each other faster than light!"

Say instead: The outcomes match, but no signal passes — the far side sees a perfect fair coin no matter what the near side does. The match only shows up when the two notebooks are brought together, and notebooks travel at ordinary speeds.

You will hear: "So the pair agreed on answers in advance, like notes in envelopes."

Say instead: Good theory — and it was tested! In choose-your-angle experiments, quantum pairs match more often than ANY pre-written notes possibly could. The experiments (Nobel Prize 2022) ruled the envelope story out. What remains is a shared blend: stranger, and proven.

PRACTICE

1. `Predict` Lab A measures FIRST in this program; lab B tallies alone afterward. Fair or tilted? Run it.

```
a = qubit
b = qubit
spin a
link a, b
say "A saw:", measure a
histogram b, shots: 300
```

2. `Read` A blog claims "quantum entanglement enables instant communication across the galaxy." Write the two-sentence correction using "fair coin" and "compare notes."
3. `Write` Build the disagreeing pair (only 01 and 10) and show BOTH: the joint tally is perfectly anti-matched, and each side alone is still a fair coin.
4. `Discuss` Perfect coordination with zero communication: name something in everyday life that even comes close. (Hard on purpose — nature's version has no everyday twin.)
5. `Design` Storyboard a 30-second myth-busting video: panel 1 the myth, panel 2 the fair-coin evidence, panel 3 the honest miracle. Captions in kid-honest language.

CHALLENGE

The saboteur test: give lab A every tool from the course — flip, turn by any angle, phase, even measuring or not measuring — and hunt for ANY choice that shifts lab B's lone 300-shot tally. Log at least three attempts with their tallies. (You are re-deriving the no-signaling rule by exhaustion — welcome to real physics.)

```
a = qubit
b = qubit
spin a
link a, b
turn a, by: 45          # saboteur's move — try others
histogram b, shots: 300
```

PROJECTOR DEMO

Correlated but silent — rehearse it once the night before; the pinned seed replays identically in class.

```
# What does the far lab actually see?
a = qubit
b = qubit
spin a
link a, b                  # b travels far away
flip a                     # lab A "signals"...
say "lab A sent:", measure a
histogram b, shots: 200    # lab B: a fair coin, regardless
```

TEACHING TIPS

- Misconception: this myth is the most durable in all of popular quantum physics, and one lesson will not fully kill it. What sticks is the METHOD — "check what the far side alone can see" — more than the conclusion. Drill the method.
- Projector move: run the three lab-B tallies (partner idle / measured / flipped) consecutively without comment, letting the sameness speak. Then ask "who detected lab A?" into the silence.
- Fast finishers: the saboteur challenge is bottomless; require logged tallies per attempt. If someone claims a hit, the class replays their seed — a perfect use of Lesson 10.

A BIT OF HISTORY

Alain Aspect, John Clauser, and Anton Zeilinger (2022) shared the Nobel Prize for Bell-test experiments spanning five decades — closing loophole after loophole until no pre-written-answers story survived. The shared blend is now measured fact.

WHY IT MATTERS (REAL WORLD)

- Quantum key distribution networks (Unit 7 builds the toy version) turn exactly this physics into eavesdrop-evident security — the no-copying, no-signaling rules doing honest work.
- China's Micius satellite distributed entangled pairs over 1,200 km — record-scale correlation, and still not one bit of signaling, exactly as the theorem demands.

CHECK FOR UNDERSTANDING

Exit ticket: (a) what lab B's lone tally shows while lab A flips and measures, (b) where the correlation becomes visible and what has to travel for that, and (c) the envelope story's fate in one sentence.

Unit 5 rubric — Two Qubits & Entanglement

SKILL	BEGINNING	DEVELOPING	SECURE
Reads joint states	Counts bars for two qubits and reads two-digit labels	Reads labels MSB-first reliably and computes corner chances for independent qubits	Predicts which corners survive a given preparation and proves it by plot and tally
Understands entanglement honestly	Builds a Bell pair and reports that outcomes always agree	States the no-story rule and shows each side alone tallies as a fair coin	Contrasts shared-blend and sealed-envelope stories and cites the experiments that decided
Holds the no-signaling line	Repeats that entanglement carries no signal	Shows by experiment that the far side's lone tally never shifts	Explains where the correlation becomes visible and dismantles a signaling scheme unaided

Edit any wording to fit your class before printing.

Lesson 17 · Repeat: loops that build big from small

Unit 6 · Programs at Scale · One 45–60 minute class.

repeat runs a block again and again — and because quantum moves compose exactly, three turns of 30 ARE one turn of 90. Loops turn small precise steps into big precise journeys.

STUDENTS WILL BE ABLE TO

- Use repeat … end to run a block a chosen number of times.
- Verify by bars that repeated turns compose exactly (3 × 30° = 90°).
- Choose loop counts to hit a target state, including overshooting past 180°.

KEY VOCABULARY

WORD	IN PLAIN WORDS	IN QUBBLE
loop	a block of instructions run several times in a row	`repeat 3 … end`
block	the indented lines between repeat and end — the part that repeats	–
compose	small moves adding into one big move, exactly — no drift, no rounding	–

BACKGROUND FOR THE TEACHER (THE NIGHT-BEFORE READ)

The first pure programming lesson since Unit 2, and it carries a physics gift: gate composition is EXACT. Three turns of 30 land precisely where one turn of 90 lands — the bars match to every decimal. Ordinary life rarely composes so cleanly; quantum gates always do.

Teach the anatomy strictly: repeat N, an indented block, end. The block can hold several lines; loops can even nest (Qubble allows three levels). Off-by-one thinking starts here — repeat 6 of turn 30 is 180, a full flip, and predicting that BEFORE running is the lesson's little victory.

The overshoot question is the fun one: repeat 9 of turn by 30 is 270 degrees — past the flip. Where do the bars land? (Same heights as 90 — chance ¾… no wait, check with your class against the bars: 270 gives the same HEIGHTS as 90, and the display's phase marking differs. Let the room discover that the journey continues around rather than bouncing back.)

Frame for the unit: this lesson is the "many steps" tool; next lesson is the "many qubits" tool; then both hit the wall that makes quantum computing genuinely hard to simulate. Loops first, because they keep programs short enough to read as the qubit count grows.

THE HOOK

Who can turn exactly 90 degrees? Easy. Who can turn exactly 90 degrees by taking three IDENTICAL smaller steps? Quantum gates can — to infinite precision, every time. Today: the loop, and the proof.

MATERIALS

- Studio access for every student or pair.
- The Lesson 6 angle-hunt chart if you kept it (today extends it with loops).
- Board protractor sketch marking 30-degree steps around to 270.

LESSON FLOW

1. The identical steps (12 min): three turns of 30 inside a repeat. Predict against Lesson 6 knowledge: 90 degrees is the fair blend. Bars: 0.71 / 0.71, exactly. Then run one turn of 90 beside it on the same seed — indistinguishable.

```
# Three small turns make one big one.
q = qubit
repeat 3
  turn q, by: 30   # 3 x 30 = 90, exactly
end
plot_state q       # the fair blend, built in steps
```

> **Predict first:** Three steps of 30. Where on the Lesson 6 curve does the qubit land, and what are the bars?

2. Anatomy drill (8 min): repeat N, block, end — rebuild the demo from memory; then two-line blocks (turn + phase inside one loop). Read a nested example aloud without running it.
3. Loop targets (15 min): pairs pick target states and hunt loop counts: land certain-1 (six 30s), land back home (twelve 30s), land the fair blend using only turns of 15. Each proof is a plot plus, where the class insists, a tally.

```
q = qubit
repeat 6
  turn q, by: 30   # 6 x 30 = 180: the full flip, assembled
end
plot_state q
say "landed:", measure q
```

> **Predict first:** Six steps of 30 is 180. What single gate from Unit 2 is this loop impersonating?

4. The overshoot (8 min): repeat 9 — 270 degrees. Predict, run, inspect: heights match 90's, phase marking tells the rest of the story. The journey wraps around; the display keeps the receipt.
5. Close (2 min): small exact steps, composed into big exact journeys. Next lesson the same discipline goes wide: many qubits at once.

HONEST LANGUAGE (WHEN YOU HEAR IT, SAY IT)

> *You will hear: "Loops make the program run faster."*
>
> **Say instead:** Loops make the program SHORTER to write — the qubit still takes every step. The quantum wins we studied come from interference, not from typing less; a loop is honest bookkeeping, not acceleration.

PRACTICE

1. `Predict` What single turn is this loop equal to, and what are the final bars?

```
q = qubit
repeat 4
  turn q, by: 45
```

```
end
plot_state q
```

2. `Read` Two programs: repeat 2 of turn by 90, versus turn by 180. A classmate claims they might differ "because loops run separately." What does gate composition say, and how would you prove it in one run each?
3. `Fix it` This loop should build the fair blend from five identical steps, but it will not run. Fix it.

```
q = qubit
repeat 5
  turn q, by: 18
plot_state q
```

4. `Write` Build certain-1 two different ways: one flip, and one loop of identical turns that never uses 180 in a single step. Prove both with plots.
5. `Design` Design a "loop dial" poster: one circle, 30-degree tick marks, and where repeat counts 1 through 12 of turn-by-30 land. Mark which counts give certain states and which give fair blends.

CHALLENGE

The spiral: put BOTH a turn and a phase inside one loop and run it with repeat 1, 2, 3, 4, plotting each. Log where the state visits. Can you find a repeat count that brings it exactly home? (Log your hunt — the journey is the point.)

```
q = qubit
repeat 2
  turn q, by: 45
  phase q
end
plot_state q
histogram q, shots: 200
```

PROJECTOR DEMO

Three turns, one journey — rehearse it once the night before; the pinned seed replays identically in class.

```
# Three small turns make one big one.
q = qubit
repeat 3
  turn q, by: 30   # 3 x 30 = 90, exactly
end
plot_state q       # the fair blend, built in steps
```

TEACHING TIPS

- Misconception: loops as approximation ("lots of small steps ≈ one big one"). The equality is EXACT — same-seed side-by-side runs are the two-minute proof. Physics that composes perfectly is rare; savor it.

- Projector move: build the repeat-6 flip live and ask the room to name the Unit 2 gate it impersonates before the final plot. The chorus of "flip!" is the composition lesson, self-administered.
- Fast finishers: the spiral challenge has real depth (turn and phase do not commute the way naive intuition expects); accept careful logs over neat answers.

A BIT OF HISTORY

Ada Lovelace (1843) wrote the first published program — for a machine that only existed on paper — and its central invention was the loop. Every repeat … end in your class descends from her Note G.

WHY IT MATTERS (REAL WORLD)

- Real quantum chips execute repeated pulse blocks constantly — calibration and error-correction cycles are loops running millions of times per second.
- Grover’s search (Lesson 13’s history) is literally a loop: the same amplify-the-right-answer block, repeated the calculated number of times.

CHECK FOR UNDERSTANDING

Exit ticket: write from memory a loop that builds the fair blend from three identical steps, state the final bars, and answer: is the loop exactly or approximately equal to one turn of 90?

Lesson 18 · Registers: a row of qubits

Unit 6 · Programs at Scale · One 45–60 minute class.

qubits N makes a row you can index — and a chain of links can pull an entire register into one all-agree pact. Structures, not single qubits, are how real programs are built.

STUDENTS WILL BE ABLE TO

- Allocate a register with qubits N and address members as reg[0], reg[1], …
- Build the three-qubit all-agree state (GHZ) with a link chain and read its two-bar joint display.
- Predict how register operations scale: what changes going from 3 to 4 qubits?

KEY VOCABULARY

WORD	IN PLAIN WORDS	IN QUBBLE
register	a named row of qubits made together	`reg = qubits 3`
index	the position number in square brackets — counting starts at 0	`reg[0]`
GHZ state	the all-agree pact: only all-zeros and all-ones survive (named for Greenberger, Horne, and Zeilinger)	–
chain	linking neighbor to neighbor so a property spreads down the whole row	–

BACKGROUND FOR THE TEACHER (THE NIGHT-BEFORE READ)

Registers are pure convenience with profound consequences: qubits 3 makes reg[0], reg[1], reg[2] in one line, and indexing starts at zero (the first quiet handshake with every programming language your students will ever meet).

The star construction is the link chain: spin the first qubit, then link it to the second, the second to the third. Each link extends the pact; the joint display collapses to two bars — 000 and 111 at 0.71 each. Three qubits, eight possible corners, six of them GONE. The class built a Bell pair yesterday; today they industrialize it.

Dwell on what the display says: this is genuine three-way entanglement — no member has a solo story, any lone member tallies as a fair coin, and the all-agree pattern lives only in the joint view. Lesson 15's rules scale up without amendment, which is itself the finding.

Set up tomorrow honestly: each added qubit DOUBLES the number of joint bars. Today the class felt 8; the register cap here is 12 (4,096 bars) and the reason is tomorrow's whole lesson. Leave the number 4,096 on the board as a cliffhanger.

THE HOOK

Yesterday, two qubits signed a pact. Today: get three qubits — or ten — to ALL agree, using a chain of the same one trick. And notice, as we go, how fast the bars multiply…

MATERIALS

- Studio access for every student or pair.
- Three volunteers for the human chain demo (agree-pact passed down a line).
- The number 4,096 written somewhere prominent, unexplained, all lesson.

LESSON FLOW

1. Meet the register (10 min): qubits 3, then plot. Eight labels, one standing bar (000 — everyone starts at zero). Drill indexing: which line touches the MIDDLE qubit? (reg[1]. Zero-counting, first contact.)

```
# A row of qubits.
reg = qubits 3       # reg[0], reg[1], reg[2] — count from 0
plot_state           # eight labels, one bar: 000
```

> **Predict first:** Three fresh qubits. How many labels does the joint display need — and how many bars have height?

2. The chain (15 min): spin the head, link down the line. Predict before the plot: which corners survive a pact where each neighbor copies the previous one's path? Two bars: 000 and 111. Verify agreement with the three-way tally.

```
# The all-agree pact.
reg = qubits 3
spin reg[0]          # open the head qubit's two paths
link reg[0], reg[1]  # the pact spreads...
link reg[1], reg[2]  # ...down the chain
plot_state           # 000 and 111 — nothing else
histogram reg[0], reg[1], reg[2], shots: 200
```

> **Predict first:** Eight corners exist. After the chain, how many survive, and which?

3. Scaling drill (10 min): pairs extend to qubits 4 with one more link. Before running: how many labels (16), how many surviving bars (2), what heights (unchanged 0.71)? The pact scales; the LABEL COUNT explodes. Connect to the number on the board.
4. Solo check (8 min): tally reg[2] alone — fair coin, exactly as Unit 5 promised. Three-way agreement, and every member alone is silent. Rules survive scaling: that is what makes them rules.
5. Close (2 min): rows, indexes, chains — and bars that double per qubit. Tomorrow: ride that doubling until the simulator itself sweats.

HONEST LANGUAGE (WHEN YOU HEAR IT, SAY IT)

> *You will hear: "With 3 qubits the register stores all eight numbers 0 through 7."*
>
> **Say instead:** The register can hold a blend with amplitudes across the eight labels — and one measurement still returns exactly ONE three-digit answer. "Holds amplitudes over eight possibilities" and "stores eight numbers" are different claims; only the first is true.

PRACTICE

1. `Predict` Spin every member instead of chaining. How many bars, what heights? (Compare Lesson 14's four-bar logic.)

```
reg = qubits 3
spin reg[0]
spin reg[1]
spin reg[2]
plot_state
```

2. `Read` A display over three qubits shows bars only at 010 and 101, equal heights. Describe the pact these qubits signed — who agrees with whom, and who disagrees?
3. `Fix it` This program wants a three-qubit all-agree pact, but it will not run. Fix it.

```
reg = qubits 3
spin reg[0]
link reg[0], reg[3]
link reg[1], reg[2]
plot_state
```

4. `Write` Build the OPPOSITES pact: reg[0] and reg[1] always agree, and reg[2] always disagrees with them (bars at 001 and 110 only). One well-placed flip after the chain does it.
5. `Discuss` The chain used neighbor-to-neighbor links only. Would linking reg[0] to BOTH others directly build the same state? What experiment settles it?

CHALLENGE

Break the chain: build the 4-qubit all-agree pact, then insert a phase on the LAST member before plotting. Do the bar heights change? Does anything change? Then design (on paper is fine) the interference experiment that would expose the difference — Unit 4 thinking at register scale.

```
reg = qubits 4
spin reg[0]
link reg[0], reg[1]
link reg[1], reg[2]
link reg[2], reg[3]
phase reg[3]         # a silent dial... still silent at this scale?
plot_state
```

PROJECTOR DEMO

The all-agree pact — rehearse it once the night before; the pinned seed replays identically in class.

```
# The all-agree pact.
reg = qubits 3
spin reg[0]          # open the head qubit's two paths
link reg[0], reg[1]  # the pact spreads...
link reg[1], reg[2]  # ...down the chain
plot_state           # 000 and 111 — nothing else
histogram reg[0], reg[1], reg[2], shots: 200
```

TEACHING TIPS

- Misconception: zero-indexing feels like a typo to newcomers ("the first qubit is reg[0]??"). Name it as a programming custom the whole world shares, drill it twice, move on — do not let it eat the physics.
- Projector move: the human chain (three students, agree-pact whispered down the line) before the code version gives the link chain a body. Then the code is just the chain, formalized.
- Fast finishers: the opposites-pact write-up has multiple correct flip placements; ask finishers to find two and argue whether they are truly the same program.

A BIT OF HISTORY

Ignacio Cirac and Peter Zoller (1995) blueprinted the first realistic quantum computer: a REGISTER of trapped ions, addressed one by one with lasers. The register-and-index picture your class used today is their design, still running in labs.

WHY IT MATTERS (REAL WORLD)

- Every quantum processor is organized as registers — chip diagrams read like reg[0..N] with a wiring map of allowed links.
- GHZ states are the working heart of quantum error correction: many-qubit agree-pacts that let machines notice when noise breaks a promise.

CHECK FOR UNDERSTANDING

Exit ticket: write from memory the three-qubit all-agree program; state the joint bars, the three-way tally pattern, and what reg[1] alone tallies as. Bonus: how many labels would qubits 5 need?

Lesson 19 · The doubling wall

Unit 6 · Programs at Scale · One 45–60 minute class.

Every added qubit doubles the joint bars: 12 qubits is 4,096, 50 is a quadrillion. The doubling is why simulators drown — and why real quantum hardware is worth building at all.

STUDENTS WILL BE ABLE TO

- Compute bar counts from qubit counts (2^N) and verify the doubling live in the display.
- Explain why doubling defeats classical simulation around 50 qubits — and why that exact wall is the opportunity.
- State honestly what a big blend does and does not buy (amplitudes over many possibilities ≠ many answers).

KEY VOCABULARY

WORD	IN PLAIN WORDS	IN QUBBLE
exponential growth	doubling with every step — the fastest-feeling growth there is	–
2 to the N	the bar count for N qubits: 2, 4, 8, 16 …	`why plot_state gets crowded`
simulation wall	around 50 qubits, no ordinary computer on Earth can hold the full set of amplitudes	–

BACKGROUND FOR THE TEACHER (THE NIGHT-BEFORE READ)

Cash the 4,096 cliffhanger. The doubling table on the board: 1 qubit → 2 bars … 10 → 1,024 … 12 → 4,096 (Quave's cap, and today the class touches it) … 50 → about 10^15 … 300 → more labels than atoms in the observable universe. Doubling starts cute and ends astronomical; let the table run past comfort.

The demo makes it physical: allocate qubits 12, spin the first six, and plot — 64 equal bars out of 4,096 labels, each 1/64. Every additional spin would double the standing bars again: 128, 256… The simulator on the projector is doing real work now; that strain IS the lesson.

Then the two honest framings, back to back. Why simulators lose: an ordinary computer must STORE every amplitude — 50 qubits is petabytes-and-up of bookkeeping for one state. Why that is the opportunity: a real quantum system IS its own state — nature does the bookkeeping natively, which is precisely Feynman's 1981 founding argument for building quantum computers at all.

Guard the honesty rail hard here, because this is where hype lives: many bars ≠ many answers. A 4,096-bar blend still yields ONE 12-digit outcome per measurement. The doubling is raw material; only interference engineering (Unit 4) turns raw material into answers. Both sentences on the wall, together.

THE HOOK

Fold a piece of paper 42 times and it reaches the Moon — nobody believes it, everybody can check it. Qubits pull the same trick: every one you add DOUBLES the bars. Today we ride the doubling until this very simulator starts to sweat.

MATERIALS

- Studio access (the demo is heavy — the projector machine should be decent).
- The doubling table drawn tall on the board, blank, to fill live.
- A sheet of paper for the folding hook — fold it as far as physically possible, then extrapolate.

LESSON FLOW

1. Fill the table (10 min): the class computes 2^N together up the board: 2, 4, 8 … 4,096 at N=12. Then keep going verbally: 20 (a million), 50 (a quadrillion), 300 (beyond atoms-in-the-universe). Doubling does not negotiate.
2. Touch the wall (15 min): the 12-qubit register, six spins, plot. 64 equal bars amid 4,096 labels; the display — and the machine — visibly working. Each further spin doubles the standing bars; let students predict 7 spins (128) and verify one step if the projector allows.

```
# 12 qubits: ride the doubling.
stage = qubits 12    # 4,096 labels — Quave's honest maximum
spin stage[0]        # 2 standing bars
spin stage[1]        # 4
spin stage[2]        # 8
spin stage[3]        # 16
spin stage[4]        # 32
spin stage[5]        # 64 equal possibilities in one blend
plot_state           # imagine 50 qubits. No computer can.
```

> **Predict first:** Six spins on twelve qubits: how many bars will stand, and how tall is each?

3. Why simulators drown (8 min): the storage argument in kid units — every amplitude must be written down somewhere; 50 qubits of "somewhere" outgrows every hard drive on Earth. Quave's 12-qubit cap is not shyness; it is honesty about the wall.
4. Why build them, then (8 min): flip the same fact over — a real quantum system does not WRITE its amplitudes, it IS them. Feynman's 1981 pitch, told straight: to simulate nature, build with nature's own parts. The wall, seen from the other side, is the doorway.
5. Close (4 min): the honesty pair, chorused: each qubit doubles the bars — and one measurement still returns one answer. Unit 7 finally builds programs where the doubling plus interference WINS.

HONEST LANGUAGE (WHEN YOU HEAR IT, SAY IT)

You will hear: "50 qubits means the computer computes a quadrillion things at once."

Say instead: Fifty qubits can hold amplitudes across a quadrillion possibilities — and a measurement returns exactly one 50-digit outcome. The blend is potential, not product; turning it into an answer takes the interference engineering, and that exists only for some problems.

You will hear: "Quave must be a weak simulator if it stops at 12 qubits."

Say instead: Twelve qubits is 4,096 amplitudes updated on every gate — comfortably real work. The cap is where honest interactivity ends in a browser tab; the world's biggest supercomputers hit their own wall near 50. The wall is the physics lesson, not a product flaw.

PRACTICE

1. `Predict` How many bars stand after this program, and how tall is each? Compute before you run.

```
stage = qubits 12
spin stage[0]
spin stage[1]
spin stage[2]
plot_state
```

2. `Read` A vendor brags: "our 300-qubit machine evaluates more states than there are atoms in the universe." Which half of the doubling-honesty pair is missing, and what question exposes it?
3. `Write` Write the smallest program whose joint display shows exactly 16 equal standing bars, and state the height each must have before you run.
4. `Discuss` Feynman argued that nature is expensive to simulate but cheap to BE. Restate his pitch in your own words using the doubling table.
5. `Design` Design the "doubling wall" board display for the classroom: paper-folding on one axis, qubit bars on the other, and the 50-qubit wall marked with what stands on each side of it.

CHALLENGE

Stress engineer: find the heaviest honest experiment this simulator handles smoothly on your machine — vary register size and spin count, log (qubits, spins, standing bars, felt speed). Where does the experience change, and does your log match the 2^N story?

```
stage = qubits 12
spin stage[0]
spin stage[1]
spin stage[2]
spin stage[3]
spin stage[4]
spin stage[5]
spin stage[6]        # 128 bars — push further at your own pace
plot_state
```

PROJECTOR DEMO

Riding the doubling — rehearse it once the night before; the pinned seed replays identically in class.

```
# 12 qubits: ride the doubling.
stage = qubits 12    # 4,096 labels — Quave's honest maximum
spin stage[0]        # 2 standing bars
spin stage[1]        # 4
spin stage[2]        # 8
```

```
spin stage[3]        # 16
spin stage[4]        # 32
spin stage[5]        # 64 equal possibilities in one blend
plot_state           # imagine 50 qubits. No computer can.
```

TEACHING TIPS

- Misconception: exponential growth reads as "very fast linear" until it suddenly does not. The paper-folding hook plus the filled table plus the straining projector is a three-hit combo; use all three.
- Projector move: narrate the machine's strain honestly ("the simulator is now updating 4,096 numbers per gate") — anthropomorphizing the struggle teaches the storage argument wordlessly.
- Fast finishers: the stress-engineer challenge produces genuinely different logs on different hardware — compare two students' walls and ask why they differ. (Their laptops differ; the physics does not.)

A BIT OF HISTORY

Richard Feynman (1981) asked the founding question — "can a classical computer simulate physics?" — showed the doubling makes it hopeless, and proposed machines built from quantum parts instead. The entire field is descended from that talk.

WHY IT MATTERS (REAL WORLD)

- The 2019 "quantum supremacy" experiment lived exactly at this wall: a 53-qubit chip ran a sampling task that strained the world's biggest classical supercomputers — the wall, demonstrated in public.
- Materials and drug researchers hit the same doubling from the other side: molecules ARE quantum states, which is why simulating them classically chokes and why chemistry is the most-cited quantum-computing application.

CHECK FOR UNDERSTANDING

Exit ticket: (a) bars for 12 qubits; (b) why 50 qubits defeats every classical computer, in one storage sentence; (c) the honesty pair — what a huge blend holds, and what one measurement returns.

Unit 6 rubric — Programs at Scale

SKILL	BEGINNING	DEVELOPING	SECURE
Composes with loops	Writes a correct repeat … end block	Predicts what repeated turns compose to and verifies exactness by bars	Chooses loop counts to hit target states, including wrap-around past 180
Commands registers	Allocates qubits N and indexes members from zero	Builds link chains and predicts surviving corners of an agree-pact	Extends pacts to new sizes and reasons about what scales and what stays fixed
Owns the doubling	States that each qubit doubles the bars	Computes 2^N bar counts and predicts standing bars after K spins	Explains both sides of the wall: why simulators drown and why that makes hardware worth building — without the many-answers fallacy

Edit any wording to fit your class before printing.

Lesson 20 · Case study: the rig-proof draw

Unit 7 · Thinking Like a Quantum Programmer · One 45–60 minute class.

A real design problem — pick a raffle winner NOBODY can rig — solved end to end with the course's own tools: blend every digit, check fairness in the bars, measure once.

STUDENTS WILL BE ABLE TO

- Design and build an 8-ticket quantum draw from a specification, not a listing.
- Verify fairness two independent ways: equal bars before, flat 800-shot tally after.
- Argue the trust story: why formulas, dice, and websites can be rigged in ways a measured blend cannot.

KEY VOCABULARY

WORD	IN PLAIN WORDS	IN QUBBLE
specification	what the program must DO, written before deciding how — the pro's starting point	–
audit	a check anyone can run themselves to confirm fairness — no trust in the operator required	–
ticket	one 3-digit outcome, 000 through 111: eight tickets from three qubits	–

BACKGROUND FOR THE TEACHER (THE NIGHT-BEFORE READ)

Unit 7 changes the classroom contract: lessons now start from a PROBLEM, and the class assembles the solution from owned tools. Resist showing the listing early — the assembly is the lesson.

The problem: the class raffle has eight tickets, and everyone distrusts everyone. The spec writes itself on the board: (1) all eight tickets equally likely, (2) provably so before the draw, (3) the draw itself unriggable even by the person running it. Classical candidates die pleasantly under cross-examination: a die (loadable), a website (who wrote it?), a volunteer "picking randomly" (hopeless — humans are terrible at random).

The quantum design falls out of the spec: eight tickets need three digits (2^3 — Unit 6 pays off immediately); each digit must be a fair honest coin (spin, Unit 2); fairness must be visible in advance (plot_state: eight equal bars — the audit IS the display); then measure each digit once (Unit 3: chance happens at measurement, nowhere earlier, no script to steal). Draw the winning ticket live.

The honesty layer distinguishes simulator from world, one more time: in Quave, the seeded draw is replayable — perfect for VERIFYING fairness (anyone reruns the audit), wrong for RUNNING tonight's actual raffle. On real quantum hardware, the same program is the genuine article: certified unriggable chance. Both sentences are the case study's finding.

THE HOOK

The class raffle is tonight. You suspect the die is loaded, the website is crooked, and — no offense — I might be too. Design a draw where NOBODY, including me, including you, can possibly cheat. You already own every tool this needs.

MATERIALS

- Studio access for every student or pair.
- Board space for the spec (numbered) and the cross-examination of classical candidates.
- Eight paper raffle tickets labeled 000–111 for the physical staging of the final draw.

LESSON FLOW

1. Spec first (10 min): elicit and board the three requirements. Then the cross-examination game: for each classical proposal from the class, find who could cheat it. (Every one falls. That is the point.)
2. Design from owned tools (15 min): eight tickets → how many digits? (Three.) Each digit fair → which gate? (spin.) Provable in advance → which window? (plot_state — eight equal bars, the audit anyone can read.) One winner → measure each digit. Assemble live, testing at each stage.

```
# A draw nobody can rig.
draw = qubits 3      # eight tickets: 000 .. 111
spin draw[0]         # blend the first digit fairly...
spin draw[1]         # ...the second...
spin draw[2]         # ...the third: all 8 bars now equal
plot_state           # THE AUDIT: check fairness with your own eyes
d0 = measure draw[0] # the draw: chance happens here, and only here
d1 = measure draw[1]
d2 = measure draw[2]
say "winning ticket:", d0, d1, d2
```

> **Predict first:** Before the audit plot: how many bars, what heights, and what would a RIGGED draw look like in this display?

3. The second audit (10 min): fairness after the fact — 800-shot tally over all three digits. Eight fractions hugging 1/8 with honest wobble (Lesson 9 skills, verbatim). Two independent audits now agree: bars before, tallies after.
4. Trust court (8 min): final cross-examination, now of OUR design: can the teacher rig it? (The program is eight lines; read them.) The student running it? (The audit is public.) The simulator's seed? (In Quave yes — name it! — which is why the REAL raffle wants real hardware; the design transfers unchanged.)
5. Close (2 min): the case-study method, named for reuse — spec, cross-examine, assemble from owned tools, audit twice, state the honest limits. Two more cases this unit; the method is the curriculum now.

HONEST LANGUAGE (WHEN YOU HEAR IT, SAY IT)

You will hear: "The quantum draw is fair because quantum stuff is mysterious."

Say instead: It is fair for two checkable reasons: the audit display shows eight equal bars BEFORE the draw, and the chance physically happens at the measurement — there is no earlier moment where an answer exists to steal or load. Mystery is not the argument; the bars are.

PRACTICE

1. `Predict` A saboteur swaps one spin for a turn by 60. What does the audit plot show, and which tickets gain? Run and confirm the audit catches it.

```
draw = qubits 3
spin draw[0]
turn draw[1], by: 60
spin draw[2]
plot_state
histogram draw[0], draw[1], draw[2], shots: 800
```

2. `Read` A classmate proposes measuring draw[0] BEFORE the audit plot. What does the audit show then, and why does the spec forbid the reordering?
3. `Write` The raffle grows to 16 tickets. Rebuild the draw and both audits. What changed, and what — because the rules scale — did not?
4. `Design` Write the one-page "trust manual" handed out with the draw: the audit anyone can run, what a rigged display would look like, and the honest simulator-vs-hardware caveat.
5. `Discuss` The draw is unriggable — but is it unLOSABLE? What legitimate complaints could a ticket holder still raise, and which are physics' problem versus procedure's problem?

CHALLENGE

The weighted-but-honest draw: the class votes that ticket 111 should win twice as often as each other ticket, and the weighting must be VISIBLE in the audit. Design it (which digits get which turns?), verify with an 800-shot tally, and write the new trust manual paragraph. (Openly unfair beats secretly unfair — that is the entire lesson, inverted.)

```
draw = qubits 3
spin draw[0]
spin draw[1]
turn draw[2], by: 109  # tune me: the audit must show the tilt
plot_state
histogram draw[0], draw[1], draw[2], shots: 800
```

PROJECTOR DEMO

The rig-proof draw — rehearse it once the night before; the pinned seed replays identically in class.

```
# A draw nobody can rig.
draw = qubits 3      # eight tickets: 000 .. 111
spin draw[0]         # blend each digit fairly
spin draw[1]
spin draw[2]         # all 8 bars now equal
plot_state           # THE AUDIT
histogram draw[0], draw[1], draw[2], shots: 800
```

TEACHING TIPS

- Misconception: "random" and "fair" as synonyms. The saboteur practice makes the difference operational — a rigged draw is still random, and the audit still catches it. Fairness is a checkable property, not a vibe.

- Projector move: stage the final draw with full theater — the audit plot held on screen for inspection, a student (not you!) pressing Run, the paper ticket handed over. Ceremony teaches that procedure is part of trust.
- Fast finishers: the weighted-draw challenge's angle (about 109 degrees puts ~2/9 on the heavy ticket… or does it?) rewards experimental tuning over formula-hunting. Demand the tally as proof, not the angle.

A BIT OF HISTORY

The NIST randomness team (2018) ran a Bell-test-certified quantum draw whose numbers were provably beyond any insider's control — then built a public randomness beacon on the same physics. Your eight-ticket raffle is their design philosophy, classroom-sized.

WHY IT MATTERS (REAL WORLD)

- National lotteries and election audits increasingly publish their randomness sources precisely because "trust me" failed historically — auditable chance is a civic technology.
- Cloudflare famously seeds part of the internet's encryption from a wall of lava lamps — the industrial version of "chance nobody can script," and a fun contrast with the quantum guarantee.

CHECK FOR UNDERSTANDING

Exit ticket: list the three spec requirements and, for each, the exact line or display in the program that satisfies it. One sentence: where does the chance physically happen, and why does that defeat rigging?

Lesson 21 · Case study: one question, not two

Unit 7 · Thinking Like a Quantum Programmer · One 45–60 minute class (a strong class may want two).

A mystery machine must be tested: fair or stuck? Classically that costs two questions — quantum interference gets the verdict in ONE. The first genuine quantum speedup, built by hand.

STUDENTS WILL BE ABLE TO

- State the mystery-machine problem and why classical testing needs two queries.
- Build the one-question tester and read the deterministic verdict from the interference.
- Explain in wave words how the answer-as-phase trick lets one query test both cases at once — honestly.

KEY VOCABULARY

WORD	IN PLAIN WORDS	IN QUBBLE
query	one use of the machine you are testing — the thing being counted and minimized	–
oracle	computer scientists' name for a mystery machine you may query but not open	–
phase kickback	the trick: prepare the answer qubit so the machine's answer lands as a PHASE on the question side	–
verdict	the single measured bit that settles the question: 1 = fair machine, 0 = stuck machine	–

BACKGROUND FOR THE TEACHER (THE NIGHT-BEFORE READ)

The crown jewel case study — the smallest real quantum algorithm (Deutsch's, 1985), and every piece of it is already owned by your class. Budget the assembly time generously; the payoff is a student saying "wait, ONE question did BOTH cases?" out loud.

The setup: a mystery box answers 0/1 questions. Either it is FAIR (answers differ for the two inputs) or STUCK (same answer regardless). Classically the proof needs both questions — ask 0, ask 1, compare. Two queries, no shortcut exists. In our program, the box is the link gate (a fair machine: output follows input); the class may not "open the box" — only wire it up and query.

The quantum tester, piece by piece: spin the question qubit — the query now carries amplitude on BOTH inputs (Unit 2). Prepare the answer qubit with flip-then-spin — this is the new trick, phase kickback: it makes the machine deposit its answers as SIGNS on the question paths rather than digits on the answer wire (Unit 4's dial, used as a notepad). One query to the box. Spin the question qubit again — paths collide, and the phases decide: answers differed → amplitudes align on $|1\rangle$, verdict certain 1. A stuck box would leave phases matched → certain 0. Run it: verdict 1, every seed, every time.

Then the honest boundary, stated with full respect: this trick tests a GLOBAL property (same-or-different) — it never learns which answer went with which input. One question bought exactly one bit, the clever part is

WHICH bit. Scaled cousins of this trick (Deutsch–Jozsa, Bernstein–Vazirani, Shor) win big for the same reason: they ask about patterns, not values. That sentence is the honest heart of all quantum speedups.

THE HOOK

A sealed machine answers yes/no questions. Is it honest or does it always say the same thing? You get to ask it exactly ONE question. Impossible? Classical logic says yes, impossible. Your Unit 4 says otherwise.

MATERIALS

- Studio access for every student or pair.
- A physical "mystery box" prop with two card inputs and an answer slot, for acting out the classical two-query proof.
- Board space for the two-paths phase trace — the lesson lives or dies on this diagram.

LESSON FLOW

1. The impossibility, felt (10 min): act out the classical game with the box prop. Ask-0-then-ask-1 works; every one-question strategy fails under class cross-examination. Write the score on the board: classical, 2 queries. Then the offer: quantum, 1.
2. Assemble the tester (15 min): build in stages with a plot after each — the blended question (spin a), the phase-notepad preparation (flip b, spin b), THE single query (link), the recombination (spin a). Then the verdict measurement. Run on three seeds: 1, 1, 1. Certain.

   ```
   # Test the machine with ONE question.
   a = qubit            # the question qubit
   b = qubit            # the answer qubit
   flip b               # prepare the notepad...
   spin b               # ...answers will land as phases
   spin a               # ask about BOTH inputs in one blend
   link a, b            # THE mystery machine — queried once
   spin a               # paths collide; phases decide
   say "verdict:", measure a   # 1 = fair machine, certain
   ```

   > **Predict first:** The box was queried once. If its two answers differ, the paths carry opposite signs into the last spin. Where does the amplitude land?

3. Trace the why (12 min): the four-number phase trace on the board, exactly like Lesson 12 but with the box in the middle. Fair box → opposite signs → collide to certain 1. Stuck box (wire b straight, no link) → matched signs → certain 0. The class traces the stuck case themselves and predicts 0 before running it.
4. The honest boundary (8 min): what did one query actually buy? Same-or-different ONLY — never which-is-which. Try to name what we DON'T know after the verdict. Then the generalization, one sentence: quantum algorithms win by asking about PATTERNS, and this eight-line program is the smallest member of the family that ends at Shor.
5. Close (5 min): score on the board updated — classical 2, quantum 1, verdict certain. The class has now built a real quantum algorithm from parts they understand line by line. Case-study method, second notch.

HONEST LANGUAGE (WHEN YOU HEAR IT, SAY IT)

You will hear: "The quantum computer asked both questions at once, so it got both answers."

Say instead: It queried in a blend of both inputs — but the verdict is ONE bit: same-or-different. We finish knowing the machine is fair yet not knowing either individual answer! The win is asking a smarter question, never getting free extra answers.

PRACTICE

1. `Predict` The STUCK machine: no link at all (the box ignores its input). Trace the phases and call the verdict before running.

```
a = qubit
b = qubit
flip b
spin b
spin a
# (the stuck machine answers without looking: no link)
spin a
say "verdict:", measure a
```

2. `Read` A classmate omits the flip-then-spin preparation and their tester returns random verdicts. Explain, in notepad language, what job the preparation was doing.
3. `Fix it` This tester should output a certain verdict, but it will not run. Fix it.

```
a = qubit
b = qubit
flip b
spin b
spin a
link a b
spin a
say "verdict:", measure a
```

4. `Write` Package the tester with a say line that prints a human-readable report before the verdict (what 1 means, what 0 means), and verify both machines against it.
5. `Discuss` The verdict tells you the machine is fair but not which input maps to which answer. Invent a real-world situation where the pattern is worth more than the values — and one where it is not.

CHALLENGE

The auditor's dilemma: your tester says "fair" with certainty — for a box you cannot open. Write the auditor's report: what exactly is certain, what is unknown, what would a cheating box-builder have to do to fool this test (can they?), and what single classical follow-up question would complete the picture. Then run that follow-up: flip a to choose input 1 explicitly, query, and read the answer digit.

```
# The classical follow-up: ask input 1, read the answer.
a = qubit
b = qubit
```

```
flip a               # choose input 1 explicitly
link a, b            # query the fair machine
say "input 1 answers:", measure b
```

PROJECTOR DEMO

One question, not two — rehearse it once the night before; the pinned seed replays identically in class.

```
# Test the machine with ONE question.
a = qubit            # the question qubit
b = qubit            # the answer qubit
flip b               # prepare the notepad...
spin b               # ...answers will land as phases
spin a               # ask about BOTH inputs in one blend
link a, b            # THE mystery machine — queried once
spin a               # paths collide; phases decide
say "verdict:", measure a   # 1 = fair machine, certain
```

TEACHING TIPS

- Misconception: the free-lunch reading ("both answers for one query"). The strongest antidote is the class's own realization that they cannot name either individual answer after the verdict. Ask them to try; let the failure teach.
- Projector move: the three-seed certainty run matters even more here than in Lesson 12 — a random-looking algorithm giving a certain verdict is the whole miracle. Certain, certain, certain.
- Fast finishers: the auditor's dilemma is genuinely deep (a cheating box that answers differently to blended vs definite queries would need to detect the difference — can it?). Accept careful partial reasoning; this question has research-grade versions.

A BIT OF HISTORY

David Deutsch (1985) published this exact algorithm — the first program in history where quantum physics provably beats every classical strategy. It saves just one query, and it opened the field: Shor and Grover are its direct descendants.

WHY IT MATTERS (REAL WORLD)

- Every quantum algorithm in industry labs today is a scaled cousin of this pattern: encode answers as phases, interfere, read a property of the whole.
- The "test a component with fewer queries" framing is real engineering — verification of chips and codes is a query-counting discipline, and quantum property-testing is an active research field.

CHECK FOR UNDERSTANDING

Exit ticket: (a) the classical query count and the quantum one; (b) the verdict bit's honest meaning — what it tells and what it withholds; (c) in one sentence, the job of the flip-then-spin preparation.

Lesson 22 · Case study: the shared secret

Unit 7 · Thinking Like a Quantum Programmer · One 45–60 minute class.

Two people need a secret key nobody else can copy. A Bell pair hands each side the same random bit — and physics itself stands guard: unriggable bits, no copying, eavesdroppers leave fingerprints.

STUDENTS WILL BE ABLE TO

- Build the one-bit key exchange: Bell pair, one measurement each, matching random bits.
- Stack the three physics guarantees: honest randomness (no script), no-cloning (no copies), measurement disturbance (no silent peeking).
- Run the repetition honestly: a multi-bit key via repeated fresh pairs, and state what Qubble cannot show (the angled eavesdropper test).

KEY VOCABULARY

WORD	IN PLAIN WORDS	IN QUBBLE
key	a shared secret string of bits that locks and unlocks messages	–
eavesdropper	someone secretly listening in the middle — the person the design must defeat	–
no-cloning rule	nature's anti-copying law: an unknown qubit cannot be duplicated, full stop	–
key rate	how many good key bits you harvest per pair — the currency of real systems	–

BACKGROUND FOR THE TEACHER (THE NIGHT-BEFORE READ)

The third case study runs the method on SECURITY, where the honesty rules earn their keep in public. The problem: two students want to pass notes nobody else can read. Any lock needs a shared key; sharing the key IS the vulnerable step. Classical key-sharing candidates fall in cross-examination: whisper it (overheard), write it (photographed), send it encrypted (with what key? — enjoy the loop).

The quantum exchange is Lesson 15's Bell pair with a job: spin, link, hand one qubit to each side, both measure. Each holds the same bit; neither chose it; it never traveled as a readable message. Repeat with fresh pairs for more bits: the demo harvests one bit and shows the joint tally as the always-agree receipt.

The security stack, told as three short course callbacks: the bits are honestly random (Lesson 3 — no script existed to steal). The qubits cannot be copied in transit (Lesson 15's anti-copying law — a wiretap that COPIES is physically impossible). And a peeking eavesdropper must MEASURE, which disturbs the state and leaves statistical fingerprints (Lesson 8 — no gentle peek exists). Detection-versus-prevention deserves a beat: quantum key exchange does not stop listening; it makes listening VISIBLE, and visible eavesdroppers mean you discard the key and start over — nothing was lost but time.

Full honesty on the boundary: catching the eavesdropper for real requires the angled measurements of the Bell test (Lesson 16's adult experiments), which Qubble does not have — say it plainly. And the simulator caveat, one last time: a seeded secret is a rehearsal; the real exchange needs real hardware, where the design transfers unchanged. Ekert published exactly this scheme in 1991; city networks run it today.

THE HOOK

You and your best friend need a secret code. But every way of sharing it — whispering, writing, texting — can be intercepted, copied, and you would NEVER KNOW. What if physics could promise: unstealable dice, uncopyable mail, and any spy leaves fingerprints? It does. You built the parts weeks ago.

MATERIALS

- Studio access for every student or pair.
- Two students staged at opposite classroom corners as the key-sharing parties.
- The classical candidates on the board for the ritual cross-examination.

LESSON FLOW

1. The loop of doom (8 min): cross-examine classical key-sharing until the class discovers the bootstrapping loop (protecting the key needs a key). Name the real problem: DISTRIBUTION, not encryption.
2. The exchange (12 min): build it live with the two corner students narrating their halves. One pair, two measurements, two matching random bits — and the joint tally as the agreement receipt. Repeat runs harvest more bits.

```
# One bit of an un-copyable key.
a = qubit            # your qubit
b = qubit            # your friend's qubit
spin a               # open the blend...
link a, b            # ...entangle: ONE shared state
mine = measure a     # your bit — random, scriptless
theirs = measure b   # your friend's bit — always matches
say "my key bit:", mine
say "friend bit:", theirs
```

> **Predict first:** Neither side chooses the bit and it never travels as a message. What guarantees they end up EQUAL?

3. The security stack (12 min): three callbacks on the board, each argued by a student who "owns" that earlier lesson: no script (L3), no copies (L15), no silent peek (L8). Then detection-vs-prevention: the fingerprints don't stop the spy; they UNMASK the spy, and unmasked spies cost you nothing but a fresh start.
4. Honest borders (8 min): two confessions, written by the class as "limitations" section of a lab report: (1) the real eavesdropper test needs angled measurements Qubble does not have — adults run Bell tests for this; (2) a seeded simulator key is a rehearsal, not a secret. Design transfers; guarantees await hardware.
5. Close (5 min): the unit's method retired with full honors — three problems, three specs, three audits, three honest-limits sections. Next unit: the machines that run these designs for real, and the grand challenge jam.

HONEST LANGUAGE (WHEN YOU HEAR IT, SAY IT)

You will hear: "The friends are sending secret messages through the entanglement."

Say instead: No message crosses — Lesson 16 stands. Each side just reads their half of a shared random state. The key is born identical at both ends; that is coordination, not communication, and it is exactly what key-sharing needs.

You will hear: "Quantum encryption is unbreakable, so the internet is safe forever."

Say instead: Quantum key EXCHANGE makes the key-sharing step eavesdrop-evident — a real, running technology. The rest of security (bugs, passwords, humans) remains gloriously breakable. One link in the chain got a physics upgrade; chains still have other links.

PRACTICE

1. `Predict` Harvest three key bits by running three fresh pairs in one program. What does each pair guarantee, and can two DIFFERENT pairs' bits disagree?

   ```
   a = qubit
   b = qubit
   spin a
   link a, b
   say measure a, measure b
   c = qubit
   d = qubit
   spin c
   link c, d
   say measure c, measure d
   ```

2. `Read` An eavesdropper intercepts b, measures it, and passes it along. Using Lessons 8 and 15, describe what the eavesdropper now holds, what b's state is, and where the fingerprints would show.
3. `Write` Write the "agreement receipt" version: the exchange plus a 200-shot joint tally proving only 00 and 11 occur. Label which lines are the key and which are the audit.
4. `Discuss` Detection instead of prevention: name another security system that works by making attacks VISIBLE rather than impossible. When is that trade-off enough?
5. `Design` Draft the protocol card two students would actually follow: how many pairs, who announces what publicly, when they discard everything, and what "fingerprints found" concretely means for tomorrow's attempt.

CHALLENGE

The disagreeing key: build the anti-matched pair (only 01 and 10) and write the two-line rule that turns perfect DISagreement into a working shared key. Is this variant any less secure? (Prove your answer with the same three-callback stack.)

```
a = qubit
b = qubit
spin a
link a, b
flip b               # now they always disagree
say "mine:", measure a
say "theirs:", measure b
histogram a, b, shots: 200
```

PROJECTOR DEMO

The shared secret — rehearse it once the night before; the pinned seed replays identically in class.

```
# One bit of an un-copyable key.
a = qubit           # your qubit
b = qubit           # your friend's qubit
spin a              # open the blend...
link a, b           # ...entangle: ONE shared state
mine = measure a    # your bit — random, scriptless
theirs = measure b  # your friend's bit — always matches
say "my key bit:", mine
say "friend bit:", theirs
```

TEACHING TIPS

- Misconception: encryption vs key exchange blur into "quantum makes it secret." Keep the surgical framing: quantum fixes the DISTRIBUTION step only, and that step was the loop of doom — a precise upgrade, honestly bounded.
- Projector move: the two corner students physically holding "their" halves while the class runs the program makes the never-traveled property visceral. Let them announce their bits simultaneously.
- Fast finishers: the disagreeing-key challenge's answer (one side flips every received bit) is one line of protocol — the interesting part is arguing security is unchanged. Demand the three-callback argument in writing.

A BIT OF HISTORY

Artur Ekert (1991) proposed key exchange built on entangled pairs — with Bell-test fingerprint-checking as the security audit. Metropolitan networks in several countries now run his idea over fiber and even satellite.

WHY IT MATTERS (REAL WORLD)

- Banks and government links in China, Switzerland, and elsewhere run quantum key distribution commercially — the shared-secret design at city scale.
- The looming Shor threat to today's encryption (Lesson 13) is why agencies are standardizing replacements NOW — quantum broke into the security world from both sides at once.

CHECK FOR UNDERSTANDING

Exit ticket: the three physics guarantees, each tagged with its lesson number and one-line argument; plus one honest limitation of running this in a simulator versus real hardware.

Unit 7 rubric — Thinking Like a Quantum Programmer

SKILL	BEGINNING	DEVELOPING	SECURE
Designs from a spec	Follows a built solution and names which tool solves which requirement	Assembles a solution from owned tools given the spec	Writes the spec, cross-examines alternatives, and defends each design choice
Audits claims with experiments	Runs a provided audit and reads it	Designs a fairness or agreement audit (bars before, tallies after) for a given program	Layers independent audits and states precisely what each one does and does not certify
States honest limits	Repeats a case study's limitation when reminded	Names what the simulator version proves versus what needs real hardware	Writes the limitations section unaided, including what one measured bit does and does not tell you

Edit any wording to fit your class before printing.

Lesson 23 · Inside a real quantum machine

Unit 8 · Real Machines & Grand Challenges · One 45–60 minute class.

Real qubits live at the bottom of golden chandeliers, colder than deep space, fighting decoherence every microsecond. Engineering, not magic — and the fight explains everything about why this is hard.

STUDENTS WILL BE ABLE TO

- Tour the anatomy of a real machine: the chandelier cryostat, the chip, the control wiring, the shielded room.
- Define decoherence as the enemy: stray heat, light, and vibration measuring your qubits before you do.
- Connect the engineering to course physics: why cold, why shielded, why error correction spends many physical qubits per useful one.

KEY VOCABULARY

WORD	IN PLAIN WORDS	IN QUBBLE
cryostat	the giant golden "chandelier" — a nested fridge cooling the chip near absolute zero	–
decoherence	the environment accidentally measuring your qubits — blends leaking away before the program finishes	–
coherence time	how long a qubit's blend survives — the machine's real budget, measured in microseconds	–
error correction	weaving many physical qubits into one tougher logical qubit using agree-pacts	–

BACKGROUND FOR THE TEACHER (THE NIGHT-BEFORE READ)

Time to meet the hardware the course has been simulating. The famous photo — the golden chandelier — is not the computer; it is the FRIDGE. Nested cooling stages hang toward a thumbnail-sized chip at the bottom, held near absolute zero, colder than anywhere in nature. The wiring spiraling down carries the microwave pulses that ARE the gates: every spin your class typed is, on real superconducting hardware, a shaped microwave chirp a few nanoseconds long.

Why the extremes? The class already owns the answer: Lesson 8 taught that any interaction that could reveal a qubit's state acts as a measurement. Heat, stray light, vibration — each is nature peeking, collapsing blends at random moments. That leak is decoherence, and it is the single enemy the whole machine is built to fight. Cold starves it; shielding blinds it; speed outruns it.

The budget framing makes it concrete: a good superconducting qubit holds its blend for around a hundred microseconds; gates cost tens of nanoseconds; so a program gets a few thousand moves before the state rots. Compare Quave: perfect qubits, infinite patience. The honest gap between simulator and machine is measured in exactly this budget.

Error correction closes the loop with a course callback that lands beautifully: the fix for fragile qubits is MORE qubits, woven into agree-pacts (Lesson 18's chains, industrialized) so that when noise breaks one member's promise, the others vote it back. Overheads run hundreds-to-thousands of physical qubits per logical one —

which is why vendors racing to "more qubits" are really racing decoherence, and why headline qubit counts and USEFUL qubit counts are different numbers. Different hardware families (trapped ions with lasers, photons in fibers, neutral atoms in optical tweezers) fight the same enemy with different weapons; the physics your class learned is the part every family shares.

THE HOOK

The most famous photo in quantum computing looks like a golden chandelier from a palace. It is a refrigerator. The computer is smaller than your thumbnail, hanging at the very bottom, at the coldest temperature in the known universe. Today: why every part of that sentence has to be true.

MATERIALS

- Large printouts or projected photos: full cryostat, chip close-up, shielded lab room.
- A real thumbnail-sized object (coin, SD card) for scale against the chandelier image.
- Board thermometer scale: room 293 K → deep space 2.7 K → the chip 0.015 K.

LESSON FLOW

1. The reveal (10 min): show the chandelier photo, collect guesses about what it is, then flip the reveal — fridge, not computer; the chip is the dot at the bottom. Walk the anatomy top to bottom: cooling stages, wiring, shielding, chip.
2. Name the enemy (12 min): derive decoherence from Lesson 8 — "what did we say ANY interaction that reveals state does?" So heat is nature measuring, stray light is nature measuring, vibration is nature measuring. Blends leak. The board thermometer shows the escape route: cold beyond deep space.
3. The budget (10 min): coherence-time arithmetic on the board: ~100 microseconds of blend, ~20 nanoseconds per gate → a few thousand moves per program. Then the contrast run: open the Studio, run any Unit 7 program, note the simulator's perfect patience. The gap IS the engineering problem.
4. More qubits to save the qubits (10 min): error correction as Lesson 18's pact, industrialized — noise breaks one member's promise, the pact votes it back. The overhead numbers (hundreds-to-thousands physical per logical) explain the qubit-count arms race honestly: headline counts race decoherence, not problems solved.
5. Close (3 min): machines exist, they are magnificent, and they are fighting for microseconds. Next lesson: what they have honestly achieved so far — and where the decades-vs-minutes headlines came from.

HONEST LANGUAGE (WHEN YOU HEAR IT, SAY IT)

> *You will hear: "Quantum computers are magic boxes that will be in phones soon."*
>
> **Say instead:** They are refrigerators the size of a room holding a chip that loses its quantum-ness in a ten-thousandth of a second — magnificent engineering fighting constant decoherence. Useful machines are coming; pocket ones are not the roadmap, and honesty about the fight is more interesting than magic.

PRACTICE

1. `Read` A photo caption calls the chandelier "the quantum processor." Write the two-sentence correction: what the golden structure actually is, and where the processor actually sits.
2. `Predict` A vendor doubles a qubit's coherence time. Using the budget arithmetic, what happens to the number of gates a program can run? What OTHER improvement would achieve the same effect?

3. `Discuss` Decoherence is "nature measuring your qubits without permission." Connect this to Lesson 8's rule about gentle peeks — why does the same physics that protects quantum keys also make quantum computers hard to build?
4. `Write` Write the "specs card" for an imaginary honest quantum computer ad: qubit count, coherence time, gates-per-program budget, and physical-vs-logical qubit count — with one sentence per line on why each number matters.
5. `Design` Design the shielding for a quantum lab in a noisy school: list five disturbance sources (heat, light, wifi, footsteps, the bell) and one defense for each. Which is hardest to defeat and why?

CHALLENGE

The decoherence simulator: Quave's qubits are perfect, but you can FAKE the enemy. Take the Bell-pair program and add a "stray measurement" of one qubit in the middle (nature peeking). Compare the joint tally with and without the peek on the same seed. Write two sentences: what the environment did to the pact, and why real machines fight so hard to prevent exactly this.

```
# Nature peeks: decoherence, faked by hand.
a = qubit
b = qubit
spin a
peek = measure a     # the environment "measures" uninvited
link a, b
histogram a, b, shots: 200   # compare with the un-peeked Bell pair
```

TEACHING TIPS

- Misconception: the chandelier IS the computer — the most common error in every popular article. Let students catch a real caption getting it wrong; caught errors stick better than corrected ones.
- Projector move: the thermometer board walk (room → space → chip) with a student physically walking a token down the scale gives the cold a narrative arc. Pause at deep space; the chip is colder. Let that sit.
- Fast finishers: the decoherence-simulator challenge is the deepest exercise here — the mid-program measurement genuinely destroys the pact's interference (tally gains 01/10 corners). Ask them to explain WHERE the pact died using Lesson 15 language.

A BIT OF HISTORY

Serge Haroche and David Wineland (2012) won the Nobel Prize for learning to control single quantum systems without destroying them — trapping atoms and photons one at a time. Every chandelier hanging in every lab is their toolbox, scaled up.

WHY IT MATTERS (REAL WORLD)

- Superconducting chips (several major tech companies), trapped ions, photonics, and neutral atoms are rival hardware families — same physics, different weapons against decoherence, genuinely undecided race.
- Cryogenics engineering for quantum labs is a booming career path — the field hires refrigerator experts as urgently as physicists.

CHECK FOR UNDERSTANDING

Exit ticket: (a) what the chandelier actually is and where the chip sits; (b) decoherence in one sentence built on Lesson 8’s rule; (c) why error correction spends many qubits per useful one — and what that means for reading qubit-count headlines.

Lesson 24 · Decades vs minutes — the honest wow

Unit 8 · Real Machines & Grand Challenges · One 45–60 minute class.

A real chip once did in minutes what a supercomputer would need millennia for — and the fine print (a contrived task, a classical counterattack, no useful output) makes the story MORE impressive as science, not less.

STUDENTS WILL BE ABLE TO

- Retell the 2019 quantum-supremacy experiment accurately: the task, the claim, the pushback, the aftermath.
- Apply the course's claim-testing tools ("what cancels?", "useful or milestone?") to headline stories.
- Place Quave honestly on the map: what a 12-qubit simulator is for, on which side of the wall it lives, and why that is the right side for learning.

KEY VOCABULARY

WORD	IN PLAIN WORDS	IN QUBBLE
quantum supremacy / advantage	the milestone where a quantum machine beats every classical computer at SOME task — any task, even a contrived one	–
sampling task	the contrived job in question: produce outputs with the right quantum randomness pattern — a task chosen because it rides the doubling wall	–
benchmark	a measuring-stick task, valuable for comparison rather than for its output	–
milestone	proof that a road exists — distinct from arriving anywhere useful yet	–

BACKGROUND FOR THE TEACHER (THE NIGHT-BEFORE READ)

The headline lesson, taught with the course's full honesty toolkit. The 2019 story, accurately: a 53-qubit superconducting chip ran a random-circuit sampling task — essentially "produce samples from a quantum blend too big to store classically" — in about 200 seconds. The team estimated ten thousand years for the best classical supercomputer. The word "supremacy" went around the world.

The fine print, told as the BEST part: rival researchers counterattacked with cleverer classical algorithms and cut "ten thousand years" down toward days — science working exactly as designed, claim and counter-claim sharpening each other. The task itself was contrived (its output useful for nothing except being hard to fake), chosen precisely because it lives on the far side of Lesson 19's doubling wall. And the machine was noisy — Lesson 23's decoherence budget, visible in the published data. None of this deflates the milestone: a physical machine touched territory no classical computer can reach. It proves the road exists; it does not claim arrival.

The class's claim-testing kit, now three tools strong, gets formalized: (1) what cancels? (Lesson 13 — is there an interference mechanism, or decoration?); (2) useful or milestone? (both are legitimate; confusing them is the sin); (3) what is the budget? (qubits, coherence, error correction — Lesson 23). Run real headlines through all three on the board.

Then place Quave on the map, with pride and precision: a 12-qubit HONEST simulator — every amplitude computed exactly, every bar true — living deliberately on the near side of the wall where full visibility is possible. Real machines have more qubits and no bar display (nature does not show amplitudes; Lesson 23 explained the fight even to keep them); Quave has perfect bars and a hard cap. Between the two sits the whole subject: the class learned on the only side where you can SEE, so that they can reason about the side where you cannot. That trade is the pedagogy of the entire course, said out loud once.

THE HOOK

In 2019, a chip the size of your thumbnail did in 200 seconds what the world's biggest supercomputer would need — the team calculated — ten thousand years for. Also true: the result was useful for nothing, rivals cut the gap to days, and the machine was noisy. Today: why the fine print makes the story BETTER.

MATERIALS

- Two or three real quantum headlines (one triumphant, one skeptical, one vendor press release).
- The doubling table from Lesson 19, redrawn, with the 53-qubit mark added.
- Board space for the three-tool claim-testing kit.

LESSON FLOW

1. The story, twice (12 min): tell the 2019 experiment once as the headline version, then once with the fine print (contrived task, classical counterattack, noise). Vote: which version is more impressive? Argue it out — the lesson's thesis is that version two wins.
2. The counterattack as hero (8 min): dwell on the rival researchers cutting the classical estimate. This is science's immune system: extraordinary claims drawing extraordinary scrutiny, both sides publishing. A field that polices its own headlines can be trusted with them.
3. The three-tool kit (12 min): formalize what-cancels / useful-or-milestone / what-budget on the board, then run the prepared headlines through all three, verdict by verdict. The vendor press release should fail at least one tool — let the class catch it themselves.
4. Quave on the map (10 min): the near-side/far-side picture: full visibility below 12 qubits, invisible amplitudes beyond 50. Ask the honest question directly: "why did we learn on a simulator?" and let the class assemble the answer — because HERE the bars can be seen, checked, and trusted, and the reasoning transfers.
5. Close (3 min): the wow, correctly sized: the road exists, the machines are climbing, the fine print is the evidence it is real. Final lesson next: the class builds its own demos and joins the gallery — teaching is the last skill.

HONEST LANGUAGE (WHEN YOU HEAR IT, SAY IT)

> *You will hear: "Quantum computers already beat normal computers, so it is over."*
>
> **Say instead:** One machine beat classical computers at one contrived benchmark, with rivals still shrinking the gap — a genuine milestone, honestly contested. Useful beats — chemistry, codes — are still being climbed toward. "The road exists" and "we have arrived" are different sentences; only the first is proven.

PRACTICE

1. `Read` Headline: "Quantum computer solves problem that would take 10,000 years — in minutes." Rewrite it so every word is honest, keeping it exciting. (It can be done — that is the exercise.)
2. `Discuss` The sampling task was chosen BECAUSE it is hard classically and useless practically. Was that a fair way to claim a milestone? Argue both sides, then pick.
3. `Predict` Apply tool 3 (the budget) to the 2019 machine: 53 noisy qubits, no error correction. What kinds of programs from this course could it run honestly, and which would decohere first?
4. `Write` Write the museum placard (80 words max) for the 2019 experiment: milestone, fine print, and why both belong on the plaque.
5. `Design` Design the front page of a newspaper that reports quantum news honestly: what three regular column headings would keep readers excited AND accurately calibrated?

CHALLENGE

The near-side flex: build the most impressive honest demo THIS simulator allows — your choice of blend, pact, or interference at up to 12 qubits — and write its museum placard: what it shows, what a real machine would add, and what the visitor should refuse to believe about it. (This is a dress rehearsal: next lesson you publish.)

```
# Dress rehearsal: your near-side showcase. Example seed —
stage = qubits 10
spin stage[0]
link stage[0], stage[1]
link stage[1], stage[2]
link stage[2], stage[3]
plot_state
histogram stage[0], stage[1], stage[2], stage[3], shots: 400
```

TEACHING TIPS

- Misconception: fine print as embarrassment. The lesson's engine is the inversion — scrutinized claims are STRONGER claims. If your class leaves thinking the counterattack made the story better, the lesson worked.
- Projector move: the two-tellings opener (headline version, then fine-print version) with a held vote between them creates the argument the whole lesson then settles.
- Fast finishers: the honest-rewrite practice is harder than it looks and worth workshopping aloud — the winning rewrites usually earn their excitement from the doubling wall itself, which needs no exaggeration.

A BIT OF HISTORY

The Google quantum team — and their rivals (2019) published the 53-qubit sampling milestone; IBM and others immediately published faster classical counter-methods. The argument itself — public, technical, unresolved for years — is the healthiest thing that ever happened to quantum headlines.

WHY IT MATTERS (REAL WORLD)

- Since 2019, supremacy-style experiments have been repeated and contested on several platforms — the wall is now a live experimental frontier, not a thought experiment.
- Funding agencies and standards bodies explicitly distinguish "milestone" from "useful advantage" in their roadmaps — the class's tool 2 is written into national policy documents.

CHECK FOR UNDERSTANDING

Exit ticket: the 2019 story in three honest sentences (claim, fine print, why it still matters), plus the three claim-testing tools from memory with one-line explanations.

Lesson 25 · Capstone: the demo jam

Unit 8 · Real Machines & Grand Challenges · Two to three class periods (design · build · jam).

The final exam is teaching: each student builds a tiny honest demo that makes one quantum idea land for a stranger, and publishes it to the gallery. Constraints breed clarity; clarity proves understanding.

STUDENTS WILL BE ABLE TO

- Design and build an original demo — at most 15 lines — that teaches exactly ONE course idea to someone who has never seen Quave.
- Pass the honesty review: every say line and comment obeys the course's language rules; every claim is backed by a bar or a tally.
- Publish (or share) the demo and teach it to an audience in under two minutes, then review a peer's demo constructively.

KEY VOCABULARY

WORD	IN PLAIN WORDS	IN QUBBLE
demo	a small program built to TEACH one idea — the gallery's currency	–
constraint	the deliberate limit (15 lines, one idea) that forces clear choices	–
remix	building your own version from someone else's published demo — the gallery's highest compliment	–
honesty review	the peer check that every word in a demo obeys the do-not-say rules	–

BACKGROUND FOR THE TEACHER (THE NIGHT-BEFORE READ)

The capstone inverts the room: students become the explainers, because teaching an idea is the strongest evidence of owning it. The product is a DEMO — not an app, not a trick: a program whose entire purpose is to make one idea land for a stranger pressing Run.

The constraint is the curriculum: at most 15 lines, exactly one idea, comments that teach, a seed pinned so the demo replays identically for every visitor, and a final line that leaves the stranger with something to try. Card-deck of viable ideas for the stuck: the cancel trick as a magic show; the fair-vs-loaded coin with audit; the pact that survives any distance; the doubling staircase; the one-question tester with a spoken-word script. Remixing a curriculum demo into something new is explicitly honorable — say so.

The honesty review is the capstone's teeth and the course's values made procedural: peers check every say string and comment against the banned-phrases wall, and every claim ("always agrees!", "certain!") must point at a bar or tally that shows it. A demo that says "the qubit is everything at once" fails review no matter how pretty — this is the do-not-say list, enforced by twelve-year-olds, which is the course working exactly as designed.

The jam itself is a gallery walk, not a contest with a podium: two-minute teach, audience tries it, one warm note and one sharp question per viewer. If publishing to the public gallery is available, publish (free, no accounts, remix-ready); a class-only share page works identically. Close the course by pointing forward: every demo in the public gallery was once someone's capstone; remixes are how the next stranger starts. The last lesson of the course is that explanations are also built from small honest parts.

THE HOOK

Final exam: no test. Instead — a stranger, the Run button, and two minutes. If they walk away honestly understanding ONE quantum idea because of fifteen lines you wrote, you pass. Build the demo that does it.

MATERIALS

- Studio access for every student; gallery or class share page for publishing.
- The banned-phrases wall, printed large, for the honesty review station.
- Design-doc half sheets: idea / audience takeaway / lines budget / seed / "try next" line.
- A jam-day route map: stations, two-minute timer, feedback slips (one warm, one sharp).

LESSON FLOW

1. Period 1 — design (15 min): pick ONE idea from the course and write the half-sheet design doc: the takeaway sentence a stranger should say afterward, the lines budget, the seed, the try-next line. Teacher signs off on scope (one idea only — cut, cut, cut).
2. Period 1 — build (30 min): draft within the 15-line constraint, seed pinned, comments teaching. The starter skeleton is a legal crutch: a working cancel-trick demo at 9 lines, leaving 6 to make it yours.

   ```
   # STARTER SKELETON — "the coin I can rig to be honest" (9 lines used)
   # A fair coin... that one invisible dial makes CERTAIN.
   q = qubit
   spin q               # fair blend — check the bars
   phase q              # the invisible dial (delete me: coin goes fair again)
   spin q               # paths collide
   plot_state q         # one bar: certainty, visible
   say "always:", measure q
   # Try next: delete the phase line and run again. Explain the difference!
   ```

   > **Predict first:** Skeleton check before customizing: what will the say line print, every run, and which single deletion turns certainty back into chance?

3. Period 2 — honesty review (20 min): peer stations. Every say and comment checked against the wall; every claim matched to its bar or tally; seed verified pinned. Reviews use the course sentence stems: "this claim needs its tally", "this phrase is on the wall — try the blend wording."
4. Period 2/3 — the jam (30+ min): gallery walk in rounds — two-minute teach, audience presses Run and tries the try-next line, feedback slips (one warm, one sharp). Publish finished demos to the gallery or class page; remixing a classmate's demo afterward is encouraged and credited.
5. Close (10 min): the course's final circle: each student says their demo's takeaway sentence aloud — one honest quantum sentence each, no banned phrases, twenty-five lessons compressed into a chorus. Point at the gallery: it grows by remix; so does understanding.

HONEST LANGUAGE (WHEN YOU HEAR IT, SAY IT)

You will hear: "My demo will say the qubit is in two places at once — strangers love that stuff."

Say instead: Strangers deserve the true version, and the true version demos BETTER: show the two bars, flip the invisible dial, make chance become certainty before their eyes. Nothing in the myth is as strange as what your fifteen honest lines actually do.

PRACTICE

1. `Design` Write three candidate takeaway sentences for three different demo ideas, then circle the one a stranger could actually verify with their own eyes in your demo. That one is your capstone.
2. `Read` Review this demo blurb against the wall: "Watch our qubit try every answer at once and always win!" List every violation and rewrite it honestly — keeping it exciting.
3. `Write` Write your demo's try-next line: one concrete edit a stranger can make (delete a line, change an angle, add a shot count) that deepens the idea. Test that the edit actually does.
4. `Fix it` A classmate's 15-line budget demo will not run. Fix it without spending extra lines.

```
# The pact, in three lines of physics.
a = qubit
b = qubit
spin a
link a, b
say "always agree:" measure a, measure b
```

5. `Discuss` Why does the 15-line constraint make demos BETTER, not worse? Answer with your own build experience — what did cutting force you to understand?

CHALLENGE

The impossible brief (optional, for glory): build a demo that teaches TWO ideas in 15 lines without muddling either — the reviewers decide if you pulled it off. Historical note: most professionals cannot. The attempt, documented honestly in comments, earns full marks even in failure.

TEACHING TIPS

- Misconception: the jam as competition. The rubric rewards a stranger's understanding, not applause — one warm note and one sharp question per viewer keeps feedback in that currency. No podium.
- Logistics move: the honesty-review station BEFORE the jam is non-negotiable sequencing — publishing then reviewing teaches the wrong order for life. Review, revise, then publish.
- Fast finishers: point them at remixing a struggling classmate's demo WITH them (credited) — capstone-level kindness, and the remix loop is the platform's actual growth mechanism; they are practicing the real thing.

A BIT OF HISTORY

Frank Oppenheimer (1969) founded the Exploratorium on one belief: people understand what they can try, break, and explain to each other. A gallery of small honest interactive demos, built by learners — his museum, your class, same idea.

WHY IT MATTERS (REAL WORLD)

- Explaining a system simply enough to demo it is a professional skill — research groups and companies employ people whose whole job is honest demonstration.
- The public gallery's remix loop mirrors how open-source software and science itself grow: publish small, honest, runnable things; let strangers build on them.

CHECK FOR UNDERSTANDING

Capstone rubric in brief (full rubric below): one idea taught and verifiable in the demo itself · 15-line and pinned-seed constraints met · every string honest (review passed) · try-next line tested · two-minute teach delivered and one peer review given. A non-working demo with an honest post-mortem outscores a slick demo with a banned phrase.

Unit 8 rubric — Real Machines & Grand Challenges

SKILL	BEGINNING	DEVELOPING	SECURE
Knows the real machines	Identifies the cryostat, chip, and their jobs	Explains decoherence from the measurement rule and reads a coherence budget	Connects error correction to agree-pacts and reads qubit-count headlines critically
Sizes the wow honestly	Retells the decades-vs-minutes story with at least one piece of fine print	Applies all three claim-testing tools to a quantum headline	Argues why scrutiny and fine print strengthen a milestone rather than weaken it
Teaches one idea (capstone)	Ships a runnable demo within the constraints	Ships a demo that passes honesty review and lands its takeaway with a peer	Ships a demo a stranger can learn from unaided, gives sharp kind review, and improves the work after feedback

Edit any wording to fit your class before printing.