QIS Key Concepts for K-12 Computer Science

Framework for K-12 Quantum Education

The world is in the midst of a second quantum revolution due to our ability to exquisitely control quantum systems and harness them for applications in quantum computing, communications, and sensing. Quantum information science (QIS) is an area of STEM that makes use of the laws of quantum physics for the storage, transmission, manipulation, processing, or measurement of information.

After the passage of the US National Quantum Initiative Act in December 2018 [1], the National Science Foundation and the White House Office of Science and Technology Policy (WHOSTP) assembled an interagency working group and subsequently facilitated a workshop titled "Key Concepts for Future Quantum Information Science Learners" that focused on identifying core concepts essential for helping pre-college students engage with QIS. The output of this workshop was intended as a starting point for future curricular and educator activities [2-4] aimed at K-12 and beyond. Helping pre-college students learn the QIS Key Concepts could effectively introduce them to the Second Quantum Revolution and inspire them to become future contributors and leaders in the growing field of QIS spanning quantum computing, communication, and sensing.

The framework for K-12 quantum education outlined here is an expansion of the original QIS Key Concepts, providing a detailed route towards including QIS topics in K-12 physics, chemistry, computer science and mathematics classes. The framework will be released in sections as it is completed for each subject.

As QIS is an emerging area of science connecting multiple disciplines, content and curricula developed to teach QIS should follow the best practices. The K-12 quantum education framework is intended to provide some scaffolding for creating future curricula and approaches to integrating QIS into physics, computer science, mathematics, and chemistry (mathematics and chemistry are not yet complete). The framework is expected to evolve over time, with input from educators and educational researchers.

Why quantum education at the K-12 level?

Starting quantum education in K-12 provides a larger, more diverse pool of students the opportunity to learn about this exciting field so that they can become the future leaders in this rapidly growing field. This is especially important because over the past century during which the first quantum revolution unfolded, the quantum-related fields have lacked gender, racial, and ethnic diversity. We must tap into the talents of students from diverse demographic groups in order to maintain our leadership in science and technology. Early introduction to quantum science can include information on applications and societal relevance, which will hopefully spark excitement and lead more students into later coursework and careers in STEM. Also, starting early with a conceptual, intuitive approach that doesn't rely on advanced mathematics will likely increase quantum awareness with more students, even those who do not pursue a career in QIS. In the long term, this will potentially improve public perception of QIS, moving it out of the weird, spooky, incomprehensible, unfamiliar realm.

What are some considerations to take into account when introducing QIS into the K-12 classroom?

As an emerging field that has traditionally been the realm of advanced undergraduate and graduate study with an aura of complexity, educators designing and delivering curriculum should keep the following in mind when integrating QIS into their classrooms.

1. Because existing materials in QIS are designed for more advanced students, the materials need to be adjusted to be age-appropriate for and build on prior knowledge of target students. As new educational research and data on implementation come in, the materials will change and improve over time.
2. Because the area may be intimidating, and there is no expectation in college that students have already learned this, motivational goals such as higher self-efficacy and a sense of belonging and identity [5-11] should be on equal footing with technical goals. Therefore, classrooms should focus on the following considerations:
    - Maintain a supportive atmosphere that encourages questions and exploration
    - Offer collaborative, exploratory activities
    - Offer a low-stakes educational setting (e.g. little time pressure without aggressive testing)
    - When relevant to the STEM subject, employ a learning cycle approach to develop models of quantum systems and phenomena, plan and carry out investigations to test their models, analyze and interpret data, obtain, evaluate and communicate their findings

QIS and CS Integration

Because of recent advances in the production of quantum computers and their associated potential to revolutionize society, quantum computing has been the subject of increasing media attention. Quantum computing leverages quantum mechanical phenomena to perform computation in new ways. Therefore, the integration of QIS into CS courses is a promising potential avenue for introducing students to QIS concepts.

Many computer science courses focus on solving computational problems, and computer science exercises need contexts in which to apply students' developing computational problem-solving skills. The challenge is finding appropriate CS exercises related to QIS that

- Involve computational solutions of the appropriate difficulty that require the skills already being taught in the course.

- Teach QIS concepts at an introductory level without requiring extensive math or physics skills.

The purpose of this document is to identify integrated activities that thread the needle between those two goals. We have identified several kinds of integrative activities:

- *Example*: The instructor using a QIS context as an example when introducing a CS topic. This does not require the students to complete an assignment; it is introduced by the instructor.
- *Activity:* Students engaging in an activity that has a relationship to QIS and CS but does not directly involve programming the computer.
- *Design*: Students create a language-inspecific design that could be used to create an implementation for a QIS-related concept, but for which a code-level implementation is not required. This could include designing storage (identifying variables and their types) or computation (identifying functions, their uses, and their interfaces).
- *Implementation*: Students write some code utilizing specific CS skills that implements something related to QIS.
- *Investigation*: Students perform an investigation that requires using or writing code that gives insight into something related to QIS.

There are limitations currently to integrating certain aspects of quantum information science. For example, programming for quantum computing is still in its infancy and there exist fewer than a dozen algorithm kernels that can be applied to real-world problems. The topic of algorithms is not directly in the K-12 QIS Key Concepts for two reasons. First, there is not yet a method for taking the kernels and adapting them for different problem domains using a programming language. Second, the level of programming and QIS experience necessary to truly understand those algorithms is more than what would reasonably fit in high school. While one can gain an intuitive understanding of a few toy algorithms that are stepping stones to useful ones, programming a quantum computer is not part of the K-12 QIS Key Concepts, and thus we do not include this in the current framework document.

QIS K-12 Key Concepts Computer Science Focus Group

The purpose of the QIS K-12 Key Concepts CS Focus Group was to create a document that would be useful to curriculum developers and teachers, providing guidance about places where high school computer science learning goals can be satisfied at the same time as content in the QIS K-12 Key Concepts.

The focus group brought together 6 experts, educators familiar with teaching and research of computer science concepts at high school and/or university levels. The members were:

Diana Franklin, University of Chicago, Chicago, IL
Michael Rogers, King College Prep, Chicago, IL
Daniel Rozanski, Pittsgrove School District, Pittsgrove, NJ
Catherine Tabor, Canutillo ISD, El Paso, TX

Tom Wong, Creighton University, Omaha, NE
Brent Yen, University of Chicago, Chicago, IL

The result is a set of activities that both teach computer science skills typically taught in high school as well as very early QIS concepts. These are not meant to be full activities or lesson plans - different instructors may use different languages, choose to go into different amounts of depth, etc. We hope, however, that by providing examples of synergistic activities, curriculum developers will be able to use these (and their own ideas) to create either individual activities or sequences of activities that build computer science skills and knowledge while, at the same time, introducing students to some basic QIS concepts.

References

1. The U.S. National Quantum Initiative: From Act to Action, C. Monroe, M. Raymer and J. Taylor, *Science* 364, 440 (2019)
2. https://www.nsf.gov/news/special_reports/announcements/051820.jsp
3. https://qis-learners.research.illinois.edu/about/
4. https://q12education.org/
5. Connecting high school physics experiences, outcome expectations, physics identity, and physics career choice: A gender study, Z. Hazari, G. Sonnert, P. M. Sadler, M. C. Shanahan, *Journal of research in science teaching* 47 (8), 978-1003 (2010)
6. High school science experiences associated to mastery orientation towards learning, K. Velez, G. Potvin, Z. Hazari, Physics Education Research Conference, Minneapolis, MN (2014)
7. Examining the impact of mathematics identity on the choice of engineering careers for male and female students, C. A. P. Cass, Z. Hazari, J. Cribbs, P. M. Sadler, G. Sonnert, Frontiers in Education Conference (FIE), F2H-1-F2H-5 (2011)
8. Examining the effect of early STEM experiences as a form of STEM capital and identity capital on STEM identity: A gender study, S. M. Cohen, Z. Hazari, J. Mahadeo, G. Sonnert, P. M. Sadler, *Science Education* 105 (6), 1126-1150 (2021)
9. Examining physics identity development through two high school interventions, H. Cheng, G. Potvin, R. Khatri, L. H. Kramer, R. M. Lock, Z. Hazari, Physics Education Research Conference (2018)
10. The importance of high school physics teachers for female students' physics identity and persistence, Z. Hazari, E. Brewe, R. M. Goertzen, T. Hodapp, *The Physics Teacher* 55 (2), 96-99 (2017)
11. Obscuring power structures in the physics classroom: Linking teacher positioning, student engagement, and physics identity development, Z. Hazari, C. Cass, C. Beattie, *Journal of Research in Science Teaching* 52 (6), 735-762 (2015)

1. QUANTUM INFORMATION SCIENCE

Key Concept:
Quantum information science (QIS) exploits quantum principles to transform how information is acquired, encoded, manipulated, and applied. Quantum information science encompasses quantum computing, quantum communication, and quantum sensing, and spurs other advances in science and technology.

a. Quantum information science employs quantum mechanics, a well-tested theory that uses the mathematics of probability, vectors, algebra, trigonometry, complex numbers, and linear transformations to describe the physical world.
b. Quantum information science combines information theory and computer science, following the laws of quantum mechanics, to process information in fundamentally new ways.
c. Quantum information science has already produced and enhanced high-impact technologies such as the Global Positioning System (GPS), which depends on the extreme precision of atomic clocks based on the quantum states of atoms.

The definition of QIS could be discussed as a precursor to any of the other concepts, or in discussions of career opportunities in the field.

2. QUANTUM STATE

Key Concept:
A quantum state is a mathematical representation of a physical system, such as an atom, and provides the basis for processing quantum information.

a. Quantum states are represented by directions or vectors in an abstract space.
b. The direction of the quantum state vector expresses the probabilities of all of the possible outcomes of a set of measurements. Quantum manipulations in the physical world follow vector operations, incorporating complex numbers and negative values. This captures a behavior of physical quantum systems that cannot be described solely by the arithmetic of probability.
c. Quantum systems are fragile. For instance, measurement almost always disturbs a quantum system in a way that cannot be ignored. This fragility influences the design of computational algorithms and communication and sensing protocols.

Learner Outcomes and Activities

a. Quantum states are represented by directions or vectors in an abstract space.

CS Activity #2a-1

- **CS Learning Outcome(s)**:
    - Students will be able to identify when an array is needed.
- **QIS Learning Outcome(s)**:
    - Students will identify that quantum states are stored as vectors (arrays).
- **Description**:
  Example: When teaching arrays and giving examples of the uses of arrays, quantum states can be used as an example of a situation requiring an array. A single qubit needs to be stored in an array rather than just a single variable because of the complexity of the state. Teachers can choose the depth to which they explain what it is about a qubit state that requires an array.
- **CS tie-ins**:
    - (ECS) Data and Information

CS Activity #2a-2

- **CS Learning Outcome(s)**:
    - Students will be able to, given a piece of data, determine how to break up that data and store it in a simple data structure such as an array or set of variables.
- **QIS Learning Outcome(s)**:
    - Students will be able to state the components of quantum state and understand

    - the relationship between that and its vector representation.
- **Description**:
  Design: Design how to store the information that expresses a quantum state. This would require students to identify what information needs to be stored and what types / structures are appropriate for their storage.
- **CS tie-ins**:
    - (ECS) Data and Information
    - (CSTA 3A-CS-01, Grades 9-10) Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects.

CS Activity #2a-3

- **CS Learning Outcome(s)**:
    - Students will be able to write code that can create and manipulate elements in an array.
- **QIS Learning Outcome(s)**:
    - Students will understand the vector structure of a quantum state.
- **Description**:
  Implementation: Store a quantum state in a list (or array) in a way that makes it possible to manipulate, normalize, and answer questions about the quantum state. Optionally, this activity can also cover complex numbers, signed and unsigned numbers.
- **CS tie-ins**:
    - (ECS) Data and Information
    - (CSP) Create Project
    - (CSTA 3A-AP-14, Grades 9-10) Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables.

CS Activity #2a-4

- **CS Learning Outcome(s)**:
    - Students will be able to pass and manipulate arrays in functions.
    - Students will understand that arrays are passed by reference, so that when a change is made to an array within a function, those changes are reflected after the function returns.
- **QIS Learning Outcome(s)**:
    - Students will understand the vector structure of a quantum state.
- **Description**:
  Implementation: Write functions that accept a quantum state as an array and perform operations that change their values (such as the example below in which they set the state).
  For example,
  ```
  void set_qustate_values(float qustate[2], float val0, float val1);
  ```
- **CS tie-ins**:
    - (ECS) Data and Information

b. **The direction of the quantum state vector expresses the probabilities of all of the possible outcomes of a set of measurements. Quantum manipulations in the physical world follow vector operations, incorporating complex numbers and negative values. This captures a behavior of physical quantum systems that cannot be described solely by the arithmetic of probability.**

CS Activity #2b-1

- **CS Learning Outcome(s)**:
    - Students will be able to pass arrays into and read values from those arrays from within functions.
- **QIS Learning Outcome(s)**:
    - Students will understand the relationship between the quantum state vector and the probability of measurement outcomes.
    - Vectors store probability magnitudes
    - Probability magnitudes must be squared to obtain the probability
- **Description**:
  Implementation: Write a function that, given a quantum state as an array as well as a position in the array (corresponding to the position in the `qustate` in which a particular outcome's information is stored), determines the probability of the measurement being that outcome.
  For example,

```
float get_probability(float qustate[2], int position)
{     float val = qustate[position]; return val*val; }
```

CS Activity #2b-2

- **CS Learning Outcome(s)**:
    - Students will be able to create nested loops to perform 2-d calculations
    - Students will be able to pass 1-d and 2-d arrays into functions
    - Students will be able to read and write values into arrays within functions
    - Students will know that any changes made to an array from within a function persist after the function returns
- **QIS Learning Outcome(s)**:
    - Students will understand that quantum operations in the real world follow vector operations.
    - Students will be able to use matrix multiply to apply gates to qubit state.
- **Description**:
  Implementation: Write a function that performs general gate operations by using matrix multiplication. The matrix multiplications can be implemented with functions by using nested loops with 2-d arrays and using the fact that arrays are pass-by-reference. For example,

```
void apply_gate(float qustate[2], float gate[2][2]);
```

CS Activity #2b-3

- **CS Learning Outcome(s)**:
    - Students will be able to write a new class that models a structure with certain shared attributes and behaviors. This requires many skills, such as:
    - Students will be able to use private/public/etc. to limit visibility of data in the class.
    - Students will be able to declare and use arrays.
    - Students will be able to store data in classes and limit their scope to instance methods.
    - Students will be able to pass 1-d and 2-d arrays into functions.
    - Students will be able to implement algorithms requiring nested loops.
- **QIS Learning Outcome(s)**:
    - Students will understand the relationship between the quantum state vector elements and the probability of measurement outcomes.
    - Students will be able to calculate the probability magnitude given the probability and vice-versa.
    - Students will understand how to use matrix multiplication with quantum state vectors to implement quantum gate operations.
- **Description**:
  Implementation: Implement a class (Python, Java, or other objected-oriented language) to hold quantum state, including storing the state in an array of complex numbers and having accessors/mutators to operate with/on the class instances.

c. Quantum systems are fragile. For instance, measurement almost always disturbs a quantum system in a way that cannot be ignored. This fragility influences the design of computational algorithms and communication and sensing protocols.

CS Activity #2c-1

- **CS Learning Outcome(s)**:
    - Students will identify the need for error correction and implement error correcting codes.
- **QIS Learning Outcome(s)**:
    - Students will understand how the fragility of quantum states make them susceptible to errors.
- **Description**:
  Activity: Make a communication scheme with whistles or keychain LED lights in Morse code and send messages to each other. Then talk about the potential mistakes that could occur and how they might be able to protect against them (using their own error correction schemes). Introduce the rectangle extra-bit parity scheme. Students could code functions that secure data (e.g. the credit card verification code) and then detect

and fix errors (depending on the level of protection). Relate this to QIS by describing the types of errors that quantum computers are susceptible to, the fact that there are 7 physical qubits to protect 1 logical qubit (as opposed to the classical version where just 1 parity bit can be used to detect a single error).

CS Activity #2c-2

- **CS Learning Outcome(s)**:
  - Students will identify algorithms as being part of or not part of a classification of algorithms that are computationally intensive to solve but computationally simple to check.
  - Students will be able to calculate the computational complexity of an algorithm
  - Students will be able to identify the value in optimizations that reduce computational complexity of the algorithm at the potential risk of incorrect results for this class of problems.
- **QIS Learning Outcome(s)**:
  - Students will understand that quantum computing algorithms sometimes produce incorrect results due to errors.
  - Students will be able to articulate that QC algorithms that may give wrong results can still be faster than classical algorithms if they reduce the computational complexity sufficiently and are fast to check. QC algorithms are
- **Description**:
  Activity: Introduce the idea of one-way functions and algorithms - operations that are easy to perform one direction but very challenging (time-consuming) to perform the other direction (e.g. factoring vs. multiplication, solving N-queens vs verifying N-queens). Have students brainstorm their own examples of such problems. Relate this to QIS as such: Some QIS algorithms are very fast but also inaccurate. The reason they have a gain is that even running those algorithms several times with verification is faster than running the accurate (classical) version.
  Implementation: Have students implement the solution and the verification code for an algorithm in this class and inspect the code to gain intuition as to what determines the runtime of large problems sizes for such problems.Relate this to QIS as such: Some QIS algorithms are very fast but also inaccurate. The reason they have a gain is that even running those algorithms several times with verification is faster than running the accurate (classical) version.

3. QUANTUM MEASUREMENT

Key Concept:
Quantum applications are designed to carefully manipulate fragile quantum systems without observation to increase the probability that the final **measurement** will provide the intended result.

a. A measurement is an interaction with the quantum system that transforms a state with multiple possible outcomes into a "collapsed" state that now has only one outcome: the measured outcome.
b. A quantum state determines the probability of the outcome of a single quantum measurement, but one outcome rarely reveals complete information about the system.
c. Repeated measurements on identically prepared quantum systems are required to determine more complete information about the state.
d. Because of the limitations of quantum measurement (providing only partial information and disturbing the system), quantum states cannot be copied or duplicated.

Learner Outcomes and Activities

a. A measurement is an interaction with the quantum system that transforms a state with multiple possible outcomes into a "collapsed" state that now has only one outcome: the measured outcome.

CS Activity #3a-1

- **CS Learning Outcome(s)**:
  - Students will understand the difference between return value and modifying state.
  - Students will be able to modify the state of arrays.
  - Students will be able to use random number generators to model randomness.
- **QIS Learning Outcome(s)**:
  - Students will be able to compute the probability of a measurement outcome.
  - Students will understand how to determine the post-measurement state given a measurement outcome.
- **Description**:
  Implementation: Write a function that calls a random number generator and includes the following
    a) Measures the quantum state, `qustate`, and returns the result as the return value
    b) Changes the state of `qustate` to collapse the superposition
    `int measure(int qustate[])`

CS Activity #3a-2

- **CS Learning Outcome(s)**:
  - Students will be able to create a program that models elements of a phenomenon or process.
  - Students will be able to create interactive data visualizations.
- **QIS Learning Outcome(s)**:
  - Students will be able to compute the probability of a measurement outcome.
  - Students will understand how to determine the post-measurement state given a measurement outcome.
- **Description**:
  Implementation: Create a Scratch project that models the superposition state visually. When the measurement button is pressed, it performs a measurement, taking it out of superposition and into the measured state.
- Optionally add another element that can change the measurement basis (e.g., H/V to +/-45).
- **CS tie-ins**:
  - (ECS) Programming unit

b. A quantum state determines the probability of the outcome of a single quantum measurement, but one outcome rarely reveals complete information about the system.

CS Activity #3b-1

- **CS Learning Outcome(s)**:
  - Students will be able to use random number generators to simulate random phenomena that exist in the real world.
- **QIS Learning Outcome(s)**:
  - Students will understand the probabilistic nature of quantum measurements.
- **Description**:
  Investigation: Investigate the relationship between probabilistic events and the number of instances of those events using weather predictions as a context. First discuss what predictions actually mean. What does a 70% chance of rain mean? If it doesn't rain, was the prediction wrong? When we perform simulations, we can see multiple possible results and draw conclusions from them. Then have students write an algorithm that predicts whether or not it will rain based on a given percent possibility. Run it once. Run it 10 times. Run it 100 times. Record the result of each trial and calculate the average result. Describe the trend as you run more trials.
- **CS tie-ins**:
  - Data
  - Simulations

    - Modeling
    - Algorithms

CS Activity #3b-2

- **CS Learning Outcome(s)**:
    - Students will be able to collect and present data to highlight relationships and support claims.
- **QIS Learning Outcome(s)**:
    - Students will understand the probabilistic nature of quantum measurements.
- **Description**:
  Activity: Collect data on 100+ days of weather predictions and record the chance of rain vs the outcome (did it rain on each day). Calculate and plot the percent chance it *did* rain vs the percent chance it was *supposed to* rain for each percentage present (which is why we want as much data as possible). How does the accuracy increase as the number of days tracked increase? Are some percentages more "accurate" than others?
- **CS tie-in**:
    - Data
    - Research

CS Activity #3b-3

- **CS Learning Outcome(s)**:
    - Students will be able to propose cause-and-effect relationships and make predictions.
- **QIS Learning Outcome(s)**:
    - Students will understand that one outcome rarely reveals complete information about the system.
- **Description**:
  Activity: Assume that students have already implemented a function, `int measure(int qustate[])`, that returns measurement outcomes and changes the quantum state `qustate`. First challenge students to write an inverse function that, given the final quantum state, can determine the state prior to measurement (they won't be able to do it). Lead a discussion about what information the result of a measurement can and cannot tell about the quantum state prior to measurement. Tie this to QIS by explaining that there does not exist a measurement of the full quantum state, so this is a realistic constraint that QIS has.

c. Repeated measurements on identically prepared quantum systems are required to determine more complete information about the state.

CS Activity #3c-1

- **CS Learning Outcome(s)**:
    - Students will be able to create loops.

    - Students will be able to use random number generators to simulate random phenomena that exist in the real world.
- **QIS Learning Outcome(s)**:
    - Students will understand how repeated measurements can give more complete information about a quantum state.
- **Description**:
  Investigation: First ask the students what the probability is of rolling a 7 with two 6-sided dice (theoretical probability). Then have them write a function with the following header:
  ```
  int roll()
  ```
  The function will roll two 6-sided dice and return their sum.
  Loop the function call 10 times and count the number of sevens that were rolled.
  Compute the experimental probability. How close is this to the theoretical probability?
  Loop the function call 1000 times and count the number of sevens that were rolled.
  Compute the experimental probability. How close is this to the theoretical probability?
  Now loop it 10,000 times. Is the result any closer to the theoretical probability?
  Discuss the relationship between the number of times rolled and the distance between actual and theoretical results.
- **CS Tie in**:
    - (ECS) Data and Information
    - (CSTA 3A-CS-01, Grades 9-10) Abstraction (using the roll function in another function to compute the probability)
    - (CSTA, 3A-IC-26, Grades 9-10) Demonstrate ways a given algorithm applies to problems across disciplines

CS Activity #3c-2

- **CS Learning Outcome(s)**:
    - Students will be able to collect and present the same data in various visual formats.
- **QIS Learning Outcome(s)**:
    - Students will understand how repeated measurements can give more complete information about a quantum state.
- **Description**:
  Investigation: Have students prepare a particular quantum state. Create visualization of data from the repeated measurements of the identically prepared systems with histograms and tables.
    a. The percentage of different outcomes
    b. The accuracy vs number of trials
- **CS tie-ins**:
    - Data

CS Activity #3c-3

- **CS Learning Outcome(s)**:

    - Students will use data to highlight or propose cause-and-effect relationships, predict outcomes, or communicate an idea.
- **QIS Learning Outcome(s)**:
    - Students will understand how repeated measurements can give more complete information about a quantum state.
- **Description**:
  Investigation: Given a coin or a die, develop an algorithm that uses repeated measurements to determine how well the coin or die is manufactured. Is the coin or die equally likely to have every value?
- **CS tie-ins**:
    - Algorithm development
    - Loops
    - Simulators

CS Activity #3c-4

- **CS Learning Outcome(s)**:
    - Students will be able to incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.
    - Students will be able to create loops.
- **QIS Learning Outcome(s)**:
    - Students will understand how repeated measurements can give more complete information about a quantum state.
- **Description**:
  Implementation: Given identically-prepared qubits (with the same unknown state) and only one public function (`measure`), write a program that measures each qubit and estimates what the original state might have been. Students are provided a function, `int measure(int qustate[])` that, given a qubit state, returns a single measurement outcome and changes the quantum state `qustate`.
  (potential code solution below)

```
int ones = 0;
int num_qubits = 100;
For (int i = 0; i <= num_qubits; i++) {
      if (qubit[i].measure() == 1) {
            ones++;
      }
}
printf(“The original state was approximately %f |0> + %f|1> with some phase.\n”, 1-sqrt(ones/num_qubits), sqrt(ones/num_qubits));
```

- **CS tie-ins**:
    - Algorithm development, loops, libraries
    - CSTA 2-AP-16 Incorporate existing code, media, and libraries into original programs, and give attribution.

- ○ CSTA 3B-AP-16 Demonstrate code reuse by creating programming solutions using libraries and APIs.

CS Activity #3c-5

- **CS Learning Outcome(s)**:
  - Students will be able to incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.
  - Students will be able to create loops.
- **QIS Learning Outcome(s)**:
  - Students will understand how repeated measurements can give more complete information about a quantum state.
- **Description**:
  Investigation: Given an object with unknown state and only three public functions: `initialize`, `run_algorithm`, and `measure`, write a program that will determine the final state of the qubit after initializing and running the algorithm. Students would write a program that repeats many times, iterating through initializing, running the algorithm, then performing the measurement.
  (partial solution code below)

```
For (int i = 0; i <= num_trials; i++) {
      qb.initialize();
      qb.algorithm();
      ones_count += qb.measure();
}
probability = ones_count / num_trials;
```

- **CS tie-ins**:
  - Data encapsulation
  - Algorithm development
  - Loops
  - Initialization (constructors)
  - Simulators

d. Because of the limitations of quantum measurement (providing only partial information and disturbing the system), quantum states cannot be copied or duplicated.

CS Activity #3d-1

- **CS Learning Outcome(s)**:
  - Students will understand the difference between variables (storage) and values.
- **QIS Learning Outcome(s)**:
  - Students will understand that classical states can be copied, in contrast to quantum states which cannot be copied or duplicated.
- **Description**:

Activity: Identify all of the places in some given program in which values get replicated. Tie to QIS by discussing the fact that quantum data may not be replicated, only moved, so that makes quantum calculations very difficult.

- **CS tie-ins**:
  - The difference between variables (storage) and values.
  - Function calls / call stack, pass by value

CS Activity #3d-2

- **CS Learning Outcome(s)**:
  - Students will be able to compare and refine multiple algorithms for the same task and determine which is the most appropriate.
  - Students will be able to decompose (break down) the steps needed to solve a problem into a precise sequence of instructions.
- **QIS Learning Outcome(s)**:
  - Students will understand that classical states can be copied, in contrast to quantum states which cannot be copied or duplicated.
  - Students will understand that the no-cloning theorem creates a challenge for designing quantum algorithms.
- **Description**:
  Implementation: Write a program that does some operation without copying or duplicating any values. Variables can be reused, and values can be moved, but values cannot be replicated. Tie to QIS by discussing the fact that quantum data may not be replicated, only moved, so that makes quantum calculations very difficult.
  For example, write a program that swaps two variables, x and y, without copying any values. This is related to the SWAP gate in quantum computing.
- **CS tie-ins**:
  - The difference between variables (storage) and values.
  - Function calls / call stack, pass by value
  - Note: The swap algorithm is taught in CSP and CSA as a standard algorithm. An alternate algorithm that does not rely on copying values is required for this activity. [For example: `x = x + y` (just adding), `y = x - y` (just subtracting), `x = x - y` (just subtracting)]

4. QUBITS

Key Concept:
The quantum bit, or qubit, is the fundamental unit of quantum information, and is encoded in a physical system, such as polarization states of light, energy states of an atom, or spin states of an electron.

a. Unlike a classical bit, each qubit can represent information in a superposition, or vector sum that incorporates two mutually exclusive quantum states.
b. At a particular moment in time, a set of n classical bits can exist in only one of 2^n possible states, but a set of n qubits can exist in a superposition of all of these states. This capability allows quantum information to be stored and processed in ways that would be difficult or impossible to do classically.
c. Multiple qubits can also be entangled, where the measurement outcome of one qubit is correlated with the measurement outcomes of the others.

Learner Outcomes and Activities

a. Unlike a classical bit, each qubit can represent information in a superposition, or vector sum that incorporates two mutually exclusive quantum states.

Note: This bullet point was covered in the Quantum State key concept (other than perhaps the term superposition)

b. At a particular moment in time, a set of n classical bits can exist in only one of 2^n possible states, but a set of n qubits can exist in a superposition of all of these states. This capability allows quantum information to be stored and processed in ways that would be difficult or impossible to do classically.

CS Activity #4b-1
- **CS Learning Outcome(s)**:
  - Students will understand binary representation and exponents.
  - Students will understand computer storage requirements.
- **QIS Learning Outcome(s)**:
  - Students will understand the capability of quantum information to be stored and processed in ways that would be difficult or impossible to do classically.
- **Description**:

Activity: After presenting some basics on how classical bits are stored and quantum qubits are stored and what they represent, pose the following question: Given n qubits, calculate how many possible basis states there are.

- **CS tie-in**:
    - Binary representation
    - Storage requirements

CS Activity #4b-2

- **CS Learning Outcome(s)**:
    - Students will be able to write a new class that models a structure with certain shared attributes and behaviors.
- **QIS Learning Outcome(s)**:
    - Students will understand how qubits store quantum information.
- **Description**:
  Implementation: Write a class, `Qubits`, that stores the state for a number of qubits in array format. Assign any number of methods that operate on that state, but the following are necessary:
  ```
  Qubits(int num_qubits);
  Int[] measureState();
  ```
  Have students discussion the following questions about the above class:
    1. Given n qubits, how long must the array in `Qubits` be?
    2. How much memory does this take?
- **CS Tie-ins**:
    - Object-oriented concepts
    - Encapsulation
    - Different coding concepts

CS Activity #4b-3

- **CS Learning Outcome(s)**:
    - Students will understand computer storage requirements.
- **QIS Learning Outcome(s)**:
    - Students will understand how qubits store and process quantum information.
- **Description**:
  Activity: Watch a video on how quantum information is stored and processed. Discuss similarities and differences between QIS storage and processing as compared to classical storage and processing.

c. Multiple qubits can also be entangled, where the measurement outcome of one qubit is correlated with the measurement outcomes of the others.

Note: See the Entanglement key concept.

5. ENTANGLEMENT

Key Concept:
Entanglement, an inseparable relationship between multiple qubits, is a key property of quantum systems necessary for obtaining a quantum advantage in most QIS applications.

a. When multiple quantum systems in superposition are entangled, their measurement outcomes are correlated. Entanglement can cause correlations that are different from what is possible in a classical system.
b. An entangled quantum system of multiple qubits cannot be described solely by specifying an individual quantum state for each qubit.
c. Quantum technologies rely on entanglement in different ways. When a fragile entangled state is maintained, a computational advantage can be realized. The extreme sensitivity of entangled states, however, can enhance sensing and communication.

<u>Learner Outcomes and Activities</u>

a. When multiple quantum systems in superposition are entangled, their measurement outcomes are correlated. Entanglement can cause correlations that are different from what is possible in a classical system.

The following two activities use analogies and manipulatives to describe what is generally happening during quantum entanglement.

CS Activity #5a-1

- **CS Learning Outcome(s)**:
  - Students will be able to use data to highlight or propose cause-and-effect relationships, predict outcomes, or communicate an idea.
- **QIS Learning Outcome(s)**:
  - Students will understand that entangled states have correlated measurement outcomes.
- **Description**:
  Activity: Perform a brainstorming activity of analogies related to entanglement. Butterfly wing beats here and has an effect there. Start general and then go more specific. In what ways are these analogies similar and dissimilar to quantum entanglement? Students will brainstorm different relationships and dependencies they know about and then talk about how similar and dissimilar those are to quantum entanglement.

CS Activity #5a-2

- **CS Learning Outcome(s)**:

    - Students will be able to use data to highlight or propose cause-and-effect relationships, predict outcomes, or communicate an idea.
- **QIS Learning Outcome(s)**:
    - Students will understand that entangled states have correlated measurement outcomes.
- **Description**:
  Activity: The teacher puts a red ball or a blue ball in a box, and puts a second matching ball in a second box. The teacher knows whether both balls are red or both are blue, but the students don't know what the colors will be until they look into the boxes.
  With quantum balls, the teacher can prepare the balls in a superposition of red and blue. When students look into the boxes, the balls will always be the same color, but no one, not even the teacher, knew the colors before the measurement (this can be accomplished by having pairs of balls that are velcroed together. The instructor does not look at the color prior to placing the balls in the boxes).
  Students will be able to, given an analogy with correlated outcomes, identify the ways in which the analogy is similar and dissimilar to quantum entanglement.
  The teacher could explain that quantum entanglement is a correlation where no one knows the outcome until the qubits are measured. However, unlike the balls in the boxes, that lack of knowledge is the fact that the outcome is not predetermined. It is not that the instructor closed their eyes, but that the ball color isn't determined until the moment of measurement.

CS Activity #5a-3

- **CS Learning Outcome(s)**:
    - Students will be able to create a program that models elements of a phenomenon or process.
    - Students will be able to use random number generators to model randomness.
- **QIS Learning Outcome(s)**:
    - Students will understand that entangled states have correlated measurement outcomes.
- **Description**:
  Implementation: Write a function that, with the use of a random number generator, given an original 2-qubit quantum state, `qustate`:
    a. Performs a single measurement of a single qubit
    b. Changes the state of the `qustate` to collapse the superposition
    c. If the qubit is entangled with another qubit, updates that second qubit in the 2-qubit quantum state given the result of the measurement of the first.

b. An entangled quantum system of multiple qubits cannot be described solely by specifying an individual quantum state for each qubit.

CS Activity #5b-1

- **CS Learning Outcome(s)**:

    - Students will be able to represent data using multiple encoding schemes.
- **QIS Learning Outcome(s)**:
    - Students will understand that an entangled quantum system of multiple qubits cannot be described solely by specifying an individual quantum state for each qubit.
- **Description**:
  Activity: Given a set of two-qubit quantum states (expressed in visual, vector, or ket notation, depending on what has been presented in the course), identify which are entangled. Alternatively, write out a two-qubit entangled quantum state.

CS Activity #5b-2
- **CS Learning Outcome(s)**:
    - Students will be able to decompose (break down) the steps needed to solve a problem into a precise sequence of instructions.
- **QIS Learning Outcome(s)**:
    - Students will understand that an entangled quantum system of multiple qubits cannot be described solely by specifying an individual quantum state for each qubit.
- **Description**:
  Implementation: Write a function that, given a 2-qubit quantum state, will determine whether or not it is entangled. The mathematical problem of determining whether a 2-qubit state is entangled can be solved with Algebra 1 material.

c. Quantum technologies rely on entanglement in different ways. When a fragile entangled state is maintained, a computational advantage can be realized. The extreme sensitivity of entangled states, however, can enhance sensing and communication.

Note: This topic will be covered in the Quantum Computing, Quantum Communication, and Quantum Sensors key concepts.

6. COHERENCE

Key Concept:
For quantum information applications to be successfully completed, fragile quantum states must be preserved, or kept **coherent**.

a. Decoherence erodes superposition and entanglement through undesired interaction with the surrounding environment. Uncontrolled radiation, including light, vibration, heat, or magnetic fields, can all cause decoherence.
b. Some types of qubits are inherently isolated, whereas others require carefully-engineered materials to maintain their coherence.
c. High decoherence rates limit the length and complexity of quantum computations; implementing methods that correct errors can mitigate this issue.

Learner Outcomes and Activities

a. Decoherence erodes superposition and entanglement through undesired interaction with the surrounding environment. Uncontrolled radiation, including light, vibration, heat, or magnetic fields, can all cause decoherence.

Note: These topics are more relevant for physics and chemistry classes.

b. Some types of qubits are inherently isolated, whereas others require carefully-engineered materials to maintain their coherence.

Note: This topic is more of an engineering and materials science research issue and not as directly relevant for a computer science class.

c. High decoherence rates limit the length and complexity of quantum computations; implementing methods that correct errors can mitigate this issue.

CS Activity #6c-1

- **CS Learning Outcome(s)**:
  - Students will be able to use random number generators to model randomness.
- **QIS Learning Outcome(s)**:

    - Students will understand how high rates of random errors can affect the functioning of a program.
- **Description**:
  Investigation: Students will investigate the effect of errors on different computational applications. Students will first write a function that, given a value, randomly introduces an error into that value given a certain error range. Then they will take a variety of applications (music clip, video clip, ASCII file, bank transaction log) and determine the severity of the errors introduced compared with the severity of the outcome of the error. These are not quantitative severities, but qualitative measures of the effects.

CS Activity #6c-2

- **CS Learning Outcome(s)**:
    - Students will understand classical error detection and correction and the idea of parity.
    - Students will be able to create 2-d arrays and nested loops.
- **QIS Learning Outcome(s)**:
    - Students will learn that methods can be implemented to correct errors that occur during computation.
- **Description**:
  Activity and Implementation: First engage students in an activity involving a deck of cards. Create a grid of playing cards (about 5x5), some face up and some face down. Make sure that when you lay them out, there are an even number of face-down cards in every row and column. Then turn your back and have someone flip one over. Then turn back around and figure out which one it is (row & column of flipped card now has an odd number of face-down cards). See if they can figure out what pattern you're looking for. Explain that the last row and column are parity values, and the real data is the grid other than the parity values.
    1. Write a function that will take in a 2-d array of values (the last column and row of which are parity bits). It finds and corrects a single error value.
       ```
       void detect_and_correct(data[ROWS][COLS]);
       ```
    2. Write a function that takes in a 2-d array of data values in `ROWS-1` rows and `COLS-1` columns. This function adds parity bits in the last row and column.
       ```
       void add_parity(data[ROWS][COLS]);
       ```

CS Activity #6c-3

- **CS Learning Outcome(s)**:
    - Classical error detection
    - Arrays
    - Functions
- **QIS Learning Outcome(s)**:
    - Students will learn that methods can be implemented to correct errors that occur during computation.
- **Description**:

Implementation: Write a function that takes in a 13 digit ISBN number passed as an int or a static array. Perform a checksum on the number and return a boolean indication whether or not the ISBN number is valid or not. Can also be done with a 10 digit ISBN number or a UPC code although the algorithms for each would be different.

- `boolean checkSum(int isbn);`
- `boolean checkSum(int[] isbn);`

7. QUANTUM COMPUTING

Key Concept:
Quantum computers, which use qubits and quantum operations, will solve certain complex computational problems more efficiently than classical computers**.**

a. Qubits can represent information compactly; more information can be stored and processed using 100 qubits than with the largest conceivable classical supercomputer.
b. Quantum data can be kept in a superposition of exponentially many classical states during processing, giving quantum computers a significant speed advantage for certain computations such as factoring large numbers (exponential speed-up) and performing searches (quadratic speed-up). However, there is no speed advantage for many other types of computations.
c. A fault-tolerant quantum computer corrects all errors that occur during quantum computation, including those arising from decoherence, but error correction requires significantly more resources than the original computation.

Learner Outcomes and Activities

a. Qubits can represent information compactly; more information can be stored and processed using 100 qubits than with the largest conceivable classical supercomputer.

CS Activity #7a-1

- **CS Learning Outcome(s)**:
  - Students will understand computer storage requirements.
  - Students will be able to organize and present collected data visually to highlight relationships and support a claim.
- **QIS Learning Outcome(s)**:
  - Students will understand how qubits store and process quantum information.
- **Description**:
  Design: Discuss how many bits are needed to represent the state of 1 qubit, 2 qubits, or n qubits, on a classical computer. The key idea is that the number of bits required goes up exponentially with the number of qubits. Develop an equation (where F is the number of bits of a single floating-point number) that calculates the number of bits required to store 1, 2, and n qubits. Assume that a floating-point number requires 64 bits for a real number.

CS Activity #7a-2

- **CS Learning Outcome(s)**:

    - Students will be able to write a new class that models a structure with certain shared attributes and behaviors.
    - Students will be able to create arrays or dynamically-allocated arrays.
- **QIS Learning Outcome(s)**:
    - Students will understand how qubits store quantum information.
- **Description**:
  Implementation: Create a class or structure that will hold the data required to model n qubits. Allocate the size based on the parameter n. Depending on how sophisticated you want to get, you can have them write functions like:
    1. Return the probability of the following measurement outcome for each qubit (n outcomes):
       ```
       float probability(int[n] measurement_outcome)
       ```
    2. Return the probability of a single qubit having the following measurement output (given no other measurements):
       ```
       float probability (int measurement_outcome, int qubit)
       ```

b. Quantum data can be kept in a superposition of exponentially many classical states during processing, giving quantum computers a significant speed advantage for certain computations such as factoring large numbers (exponential speed-up) and performing searches (quadratic speed-up). However, there is no speed advantage for many other types of computations.

CS Activity #7b-1

- **CS Learning Outcome(s)**:
    - Students will be able to compare tradeoffs associated with computing technologies.
- **QIS Learning Outcome(s)**:
    - Students will understand how quantum computers can have a computational advantage over classical computers.
- **Description**:
  Activity: Invite students to reason through the prisoner's dilemma. Discuss the similarities/differences between the outcome of the prisoner's dilemma when completed using a classical algorithm and when completed using a quantum algorithm (the introduction of entanglement).
- **CS tie-ins**:
    - Appropriate for: Game Theory, Discrete Math, AP CSP
    - CSTA: 2-IC-20 Compare tradeoffs associated with computing technologies.

CS Activity #7b-2

- **CS Learning Outcome(s)**:
    - Students will understand the concept of computational complexity.

    - Students will be able to use data to highlight or propose cause-and-effect relationships, predict outcomes, or communicate an idea.
- **QIS Learning Outcome(s)**:
    - Students will understand how quantum computers can have a computational advantage over classical computers.
- **Description**:
  Investigation: Introduce a basic quantum algorithm portrayed as a game, like Deutsch's algorithm. Teach the algorithm conceptually as a game where behind a door is either a tiger or a bag of money or something like that, etc. https://arxiv.org/abs/2005.07874. Have the students implement this game classically and analyze the computational complexity (and space requirements) of their own solutions. After that, explain that there exists a quantum algorithm that can solve this in X time (with Y qubits).

CS Activity #7b-3

- **CS Learning Outcome(s)**:
    - Students will understand the concept of computational complexity.
    - Students will understand the difference in run-time between some recursive solutions and iterative solutions.
- **QIS Learning Outcome(s)**:
    - Students will understand how quantum computers can have a computational advantage over classical computers.
- **Description**:
  Investigation: With students, work on the iterative and recursive solutions to calculating the nth value in the Fibonacci sequence (or work through two different sorting algorithms). Have students run the function on increasingly large numbers, plotting the input number with the runtime. Optionally, can formally present computational complexity to calculate the complexity of each algorithm.
- **Quantum tie-in**:
  Choose a problem that has a computational complexity advantage in quantum. Present the complexity of each and state that because of quantum's unique features, there are quantum algorithms that will perform the work in fewer steps. Also mention that the clock rate for quantum computers is significantly slower, so reiterate what computational complexity is good for - comparing very large problems in which the constants don't matter, not small problems that are solved quickly already on a slow computer.

CS Activity #7b-4

- **CS Learning Outcome(s)**:
    - Students will understand that different computational techniques are appropriate for different applications (AP CS A)
- **QIS Learning Outcome(s)**:
    - Students will understand how quantum computers can have a computational advantage over classical computers.
- **Description**:

Activity: Given a list of applications, have students identify what techniques they should use. The techniques can vary: arrays vs linked lists, recursion vs iteration, quantum vs classical. The idea is to get a problem, identify its characteristics, and decide what techniques are appropriate.
Present the classes of problems that are appropriate for each:

a) arrays vs linked list (more static vs more dynamic data)
b) recursion vs iteration (tree-shaped computation vs linear computation)
c) quantum vs classical (searches, factoring, molecular simulation vs database storage, streaming, simple calculations).

Make a set of cards that have different pictures and descriptions of different applications. The cards are marked as to what they are supposed to categorize about it. Do different rounds, with students first only categorizing arrays vs linked lists. Students are in small groups (2-3 students), and they need to discuss each one. They each make three piles - array, linked list, disagree or unsure. Then go through each and vote, discussing any disagree / unsure categorizations. This isn't necessarily about being right or wrong, it's about holding things up for discussion when they aren't 100% clear so you can understand the decision criteria better. (Based on Is It Alive? from science curricula)

CS Activity #7b-5

- **CS Learning Outcome(s)**:
    - Students will understand the concept of computational complexity.
    - Students will be able to organize and present collected data visually to highlight relationships and support a claim.
- **QIS Learning Outcome(s)**:
    - Students will understand how quantum computers can have a computational advantage over classical computers.
- **Description**:
  Investigation: Have students write algorithms for prime factoring an integer n. (Using, for example, the Sieve of Eratosthenes)
  Record and graph the time needed to factor as n increases. Talk about the exponential time needed to factor numbers. The fact that factoring is time-consuming and difficult for classical computers is the basis for widely-used public key cryptography. Quantum computers are useful because they can factor numbers much faster. Present the complexity of Shor's algorithm (but not the algorithm itself) and have them graph that curve (starting at a higher spot since quantum computers will be slower on small problems) and compare to what they graphed.

c. A fault-tolerant quantum computer corrects all errors that occur during quantum computation, including those arising from decoherence, but error correction requires significantly more resources than the original computation.

Note: See the Coherence key concept for error correction activities.

8. QUANTUM COMMUNICATION

Key Concept:
Quantum communication uses entanglement or a transmission channel, such as optical fiber, to transfer quantum information between different locations.

a. Quantum teleportation is a protocol that uses entanglement to destroy quantum information at one location and recreate it at a second site, without transferring physical qubits.
b. Quantum cryptography enhances privacy based on quantum physics principles and cannot be circumvented. Due to the fragility of quantum systems, an eavesdropper's interloping measurement will almost always be detected.

Learner Outcomes and Activities

a. Quantum teleportation is a protocol that uses entanglement to destroy quantum information at one location and recreate it at a second site, without transferring physical qubits.

CS Activity #8a-1

- **CS Learning Outcome(s)**:
  - Students will be able to model processes by creating and following algorithms (sets of step-by-step instructions) to complete tasks.
  - Students will be able to model the role of protocols in transmitting data across networks and the Internet.
- **QIS Learning Outcome(s)**:
  - Students will be able to describe the quantum teleportation protocol.
- **Description**:
  Activity: Have students role-play the teleportation protocol with students playing the role of Alice and Bob. Talk about the concepts of entanglement, measurement, no-cloning involved in the teleportation protocol. Teleportation allows transfer of a qubit, but students should know that faster-than-light communication is not possible through this process.
- **CS tie-in**:
  - Emerging ideas and technology, innovations in computing technology

b. Quantum cryptography enhances privacy based on quantum physics principles and cannot be circumvented. Due to the fragility of

quantum systems, an eavesdropper's interloping measurement will almost always be detected.

CS Activity #8b-1

- **CS Learning Outcome(s)**:
  - Students will be able to model processes by creating and following algorithms (sets of step-by-step instructions) to complete tasks.
  - Students will be able to model the role of protocols in transmitting data across networks and the Internet.
- **QIS Learning Outcome(s)**:
  - Students will be able to describe the BB84 quantum key distribution algorithm.
- **Description**:
  Activity: Have students role play the BB84 quantum key distribution algorithm. Students can play the roles of the sender Alice, the receiver Bob, and the eavesdropper Eve.

CS Activity #8b-2

- **CS Learning Outcome(s)**:
  - Students will discuss real-world cybersecurity problems and how personal information can be protected.
- **QIS Learning Outcome(s)**:
  - Students will understand how the BB84 quantum key distribution allows for privacy through private key cryptography.
- **Description**:
  Activity: Show students a few private key cryptography protocols, such as XOR of key with the message. Provide a few messages for students to encode and decode, first with known starting points and answers, then allow them to send messages to each other. Relate to quantum computing by discussing the issues of distributing the secret key and why quantum cryptography helps enhance security for this type of encryption.
- **CS tie-in**:
  - 1B-NI-05 Cybersecurity 3.1

CS Activity #8b-3

- **CS Learning Outcome(s)**:
  - Students will understand how encryption algorithms work and their relationship to computing technology.
- **QIS Learning Outcome(s)**:
  - Students will understand the impact of quantum key distribution on society.
- **Description**:
  Activity: Present the challenge of distributing a secure key over insecure channels. Have students brainstorm different ideas, and draw upon spy movies (e.g. Mission Impossible and 007) and real historical contexts for realistic and amusing examples. Present quantum algorithms, including the proven security of BB84, and how this is stronger than classical methods.
- **CS tie-in**:

- Students will be able to describe the impact of a computing innovation. (5.1 Beneficial and Harmful Effects IOC-1.A, IOC-1.B 5.C)
- Students will be able to evaluate the use of computing based on legal and ethical factors. (5.E)
- Students will be able to compare how people live and work before and after the implementation or adoption of new computing technology. (1A-IC-16)
- Students will be able to recommend security measures to address various scenarios based on factors such as efficiency, feasibility, and ethical impacts. (CSTA standards: 3A-NI-06)

9. QUANTUM SENSORS

Key Concept:
Quantum sensing uses quantum states to detect and measure physical properties with the highest precision allowed by quantum mechanics.

a. The Heisenberg uncertainty principle describes a fundamental limit in simultaneously measuring two specific, separate attributes. “Squeezing” deliberately sacrifices the certainty of measuring one attribute in order to achieve higher precision in measuring the other attribute; for example, squeezing is used in LIGO to improve sensitivity to gravitational waves.
b. Quantum sensors take advantage of the fact that physical qubits are extremely sensitive to their surroundings. The same fragility that leads to rapid decoherence enables precise sensors. Examples include magnetometers, single-photon detectors, and atomic clocks for improvements in medical imaging and navigation, position, and timing.
c. Quantum sensing has vastly improved the precision and accuracy of measurements of fundamental constants, freeing the International System of Units from its dependence on one-of-a-kind artifacts. Measurement units are now defined through these fundamental constants, like the speed of light and Planck’s constant.

<u>Learner Outcomes and Activities</u>

a. The Heisenberg uncertainty principle describes a fundamental limit in simultaneously measuring two specific, separate attributes. “Squeezing” deliberately sacrifices the certainty of measuring one attribute in order to achieve higher precision in measuring the other attribute; for example, squeezing is used in LIGO to improve sensitivity to gravitational waves.

Note: There are some computer science concepts that deal with the issue of tradeoffs, e.g., time vs. space resource tradeoffs. However, high school computer science classes do not typically teach about time vs. space tradeoffs or parallel processing. These issues are also not that closely related to the tradeoffs described by the Heisenberg uncertainty principle that might be more relevant in a physics class.

b. Quantum sensors take advantage of the fact that physical qubits are extremely sensitive to their surroundings. The same fragility that leads to rapid decoherence enables precise sensors. Examples include magnetometers, single-photon detectors, and atomic clocks

for improvements in medical imaging and navigation, position, and timing.

Note: This topic provides potential context for discussion in a computer science class. For example, computers require internal clocks for time synchronization, and this key concept on quantum sensors talks about using atomic clocks for keeping time. However, computer hardware is not typically addressed in high school computer science programs and would belong to another class or program if at all in high school. In addition, different types of sensors collect information for storing and processing on computers. This provides a connection between sensors and computing that could provide context for discussion in a computer science class. However, while some high school computer science teachers might have access to sensors, this might be more appropriate in a robotics or design class, rather than a traditional computer science class in which sensors are not prescribed by any standards.

c. **Quantum sensing has vastly improved the precision and accuracy of measurements of fundamental constants, freeing the International System of Units from its dependence on one-of-a-kind artifacts. Measurement units are now defined through these fundamental constants, like the speed of light and Planck's constant.**

Note: The topic of units and fundamental constants belongs more in a physics or chemistry classroom.