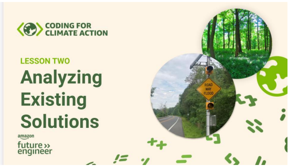
UNIT OVERVIEW Mitigating the Effects of Natural Hazards

See Full Unit Folder

Anchoring Phenomenon: Worldwide, between 1970-1979, there were 711 climate-related natural hazards and between 2010-2022 there were 3,165 climate-related natural hazards. (Sources: World Meteorological Organization and Our World in Data)

Design Problem: How can we design early warning systems to increase community resilience and mitigate the effects of natural hazards caused by climate change?

Lesson Title and Question	Lesson Snapshot	What Students will Figure Out
LESSON ONE **Define the Problem** *How can we design early warning systems to increase community resilience and mitigate the effects of natural hazards?* **1 45-minute class period** 	Students are introduced to the problem of natural hazards increasing worldwide over time due to climate change. Students analyze data to figure out that increased temperatures lead to increased evaporation rates and increased water vapor in the air. Students develop an understanding of how each natural hazard connects to climate change and then consider how Early Warning Systems can protect people from natural hazards. Students generate questions about how early warning systems work and consider obstacles to access.	**Students figure out:** ● Climate change is affecting the amount of rain (precipitation) in different parts of the U.S. (and world). ● Too much rain is one cause of some natural hazards (flooding, hurricanes, landslides). ● Not enough rain is one cause of some natural hazards (wildfires, drought). ● An Early Warning System (EWS) is a type of technology that can help protect people and save lives by mitigating the effect of natural disasters. ● Currently, not everyone has equal access to early warning systems.

Navigation to Lesson 2: We have a lot of questions about the parts of an EWS. Let's take a look at some existing solutions to learn more about them.		
LESSON TWO **Analyzing Existing Solutions** *What types of Early Warning Systems already exist?* **1 45-minute class period** 	In this lesson, students will be introduced to computational thinking and the engineering design process. Both will be used to help guide our research, design, build, testing, and sharing the results of our Early Warning System design solutions. Students will spend this entire lesson in their Computational Thinking and Engineering Journals as the teacher presents information on three existing Early Warning Systems. Students will analyze the information presented and models provided to determine the common parts and functions of all Early Warning Systems. This will begin our research phase of the engineering design process while also using the computational thinking concepts of evaluation and looking for patterns.	**Students figure out:** ● Understanding more about Early Warning Systems (EWSs) will help us design a solution to help protect everyone in a community. ● EWSs have similar parts and functions including sensors to sense the danger, a way to send/process data (cloud computing, sent over Internet), and a way to send communication to the end user (phones, loud speakers). ● EWSs require computer programmers and engineers to build the system. ● EWSs can be improved to help even more people. ● We can use computational thinking concepts to analyze Early Warning Systems.
Navigation to Lesson 3: Many of the questions and ideas you shared are related to the inputs and outputs of an early warning system so let's explore that next!		
LESSON THREE **Researching micro:bit Inputs and Outputs** *How could we use the micro:bit to design an early warning system and increase accessibility?* **2 45-minute class periods** 	Students will be provided a real world example of a warning sign that uses flashing lights to help warn people that a road is flooded. This leads into a three part design challenge where students will learn the basic logic and structure of creating code sequences for the micro:bit to create a flashing sign. Students will learn the difference between hardware and software and will test their code (software) by downloading to the micro:bit (hardware). They will experience the efficiency of using the forever loop to repeat code. Students will also learn how to add music to increase accessibility and code button inputs to display LED lights and sound outputs. Students will then reflect in their journals on the pros and cons (accessibility) of different outputs and on the computational thinking skills used in this lesson.	**Students figure out:** ● Computer science will help us design our Early Warning System. ● Outputs such as LED lights and sound can be used as an alert system. ● Using more than one type of output will increase the accessibility of the EWS.

Navigation to Lesson 4: We have just begun to learn how to code the micro:bit. Looking back at our questions, it seems like we might want to learn more about different sensors that we could use to design our early warning system.		
LESSON FOUR **Researching micro:bit Sensors** *How can we build a system that continually monitors the environment to detect danger and warn communities?* **3 45-minute class periods** 	In Part 1 of this lesson, students will learn how to fix the problem experienced with the code in Lesson 3. They will begin by creating and setting variables to store information that changes in an algorithm. In this case, they are setting the variable to Danger = 1 when danger is detected and setting the variable to Danger = 0 when no danger is detected. They will also be introduced to the Logic Toolbox drawer. In Part 2 of this lesson, students will use conditional blocks to detect the danger and trigger the variable to run the code using the input buttons. They will learn how to write comments to help explain complex algorithms. Students will then revisit sensors and begin the first Monitoring the Environment Design Challenge that uses the accelerometer to detect motion. In Part 3 of this lesson, students will continue with their first Sensor Design Challenge and will download and test with the micro:bit. Students will then use the logic comparison blocks with sensors to tell the computer to check for a certain condition in the second and third Sensor Design Challenge, using the light and temperature sensors.	**Students figure out:** ● Sensors measure inputs (signals) from the environment (light energy, gravitational pull, temperature) and convert into data for the computer to read. ● Computers use data to trigger outputs when conditions are met. ● Sensors are useful to help us design an Early Warning System in order to measure changes in the environment related to natural hazards.

Navigation to Lesson 5: *After reflection, students determine they still need something that could measure flooding and droughts. How could we use the microbit to monitor those hazards?*

[LESSON FIVE](#)
[Researching micro:bit Pins](#)

How can we measure water levels in the environment using micro:bits?

1 45-minute class periods

In this lesson, students will read about a real world example of an Early Warning System that monitors water levels called StormSense. Students then learn how to use the input pins on the micro:bit to measure when electricity completes an electric circuit using the pin pressed blocks of code. Students create a conductivity tester to figure out whether certain materials are conductors or insulators. They then build a program to use the pins to measure water levels through the detection of an electric circuit. A teacher demonstration may be done with water to show how water acts as a conductor to transfer electricity. Students will build the program to measure water levels but will use aluminum foil to represent water as the conductor. If time provides, students will engage in the extension activity that asks them to build a multi-tiered system.

Students figure out:

- Pins can be used to trigger an event through the detection of a closed circuit.
- Computer science is powered by electricity.
- We can use circuits to sense things in our environment.

Navigation to Lesson 6: *We've chosen an area to focus on and figured out which type of natural hazard is most likely to happen there. What sensors could you use to monitor natural hazards in the area you chose?*

[LESSON SIX](#)
[Design and Build Your EWS](#)

How can we use computational thinking approaches to help build our EWS design solution?

3 45-minute class periods

In this lesson, students will spend two days designing and building their Early Warning System design solutions. On Day 1 (Part 1), students will brainstorm design decisions as a group and draw a model for the type of system they decide to build. On Day 2 (Part 2), groups of students will divide themselves to split up the roles of software developers and engineers. The software developers will write the code, test with the simulator, and download to the micro:bit. The engineers will build the physical prototypes using the micro:bit kits and any consumables provided. Together they will test and troubleshoot their design solutions and document them in their journals.

Students figure out:

- Developing a model helps plan for the hardware components (inputs and outputs) and code (information flow) that will be needed to build the Early Warning System design solution.
- Building and testing a prototype of a design solution helps to identify issues with the hardware or software.
- Computational thinking approaches help to iteratively design, test, and improve a design solution.

Navigation to Lesson 7: *We've designed an EWS; how could we explain how it works to our class?*		
LESSON SEVEN **Sharing Your Design Solution** *How will you present your design solution to the class?* **1 45-minute class period** 	In this lesson, students will prepare to share their final design solutions with the class. They will review the EWS rubric as a class and add any final images to the Design Solution Slideshow or other methods for sharing their design solutions with the class. Students will then present their design solutions as a team while the other students in the class take notes in their Computational Thinking and Engineering Journals of the ideas that they think the team did well. Students will share what they think each team did well and will answer the final question in their journal, explaining any improvements they would make to their design after listening to the other presentations.	**Students figure out:** ● Computational thinking approaches help to iteratively design, test, and improve a design solution. ● For an early warning system to be effective/adopted, a team needs to accurately explain their design solution including how the software (inputs, outputs, information flow) and hardware components (sensors, micro:bit, speakers) function together in the system.

Standards

Next Generation Science Standards	CSTA Computer Science Standards
Disciplinary Core Ideas **ESS3.B Natural Hazards** • A variety of natural hazards result from natural processes. Humans cannot eliminate natural hazards but can take steps to reduce their impacts. **ETS1.B: Developing Possible Solutions** • At whatever stage, communicating with peers about proposed solutions is an important part of the design process, and shared ideas can lead to improved designs. (3-5-ETS1-2). **These lessons could be part of a series of lessons building toward the following Performance Expectation:** ***3-ESS3-1.*** *Make a claim about the merit of a design solution that reduces the impacts of a weather-related hazard.* [Clarification Statement: Examples of design solutions to weather-related hazards could include barriers to prevent flooding, wind resistant roofs, and lightning rods.]*	**1A-CS-02** Use appropriate terminology in identifying and describing the function of common physical components of computing systems (hardware). **1A-CS-01** Select and operate appropriate software to perform a variety of tasks, and recognize that users have different needs and preferences for the technology they use. **1A-AP-10** Develop programs with sequences and simple loops, to express ideas or address a problem. **1A-AP-14** Debug (identify and fix) errors in an algorithm or program that includes sequences and simple loops. **1B-AP-09** Create programs that use variables to store and modify data. **1B-AP-10** Create programs that include sequences, events, loops, and conditionals.