LESSON GOALS AND STANDARDS CROSSWALK

HOW IS COMPUTER SCIENCE CHANGING THE WORLD?

LESSON 1: ROBOTICS (featuring Ayanna Howard)

LEARNING GOALS	DESCRIPTION	WHEN MIGHT YOU USE THIS LESSON?
Students will be able to... • Define the term "robot." • Name 3 ways robots are used today. • Name 1 positive and 1 negative way robots are affecting the world.	Students learn what a robot is, read and talk about real-world examples of different kinds of robots, and have a discussion weighing the positive and negative impacts of robots on the world.	• To provide practice reading and discussing nonfiction texts. • To provide practice deconstructing and evaluating the arguments of others.
CSTA STANDARDS	**COMMON CORE ELA STANDARDS**	**COMMON CORE MATH STANDARDS**
2-IC-20: Compare tradeoffs associated with computing technologies that affect people's everyday activities and career options.	**CCRA.R.2:** Determine central ideas or themes of a text and analyze their development; summarize the key supporting details and ideas.	**MP3:** Construct viable arguments and critique the reasoning of others.

HOW DO I ALREADY THINK LIKE A COMPUTER SCIENTIST?

LESSON 2: DECOMPOSITION (featuring Ada Lovelace)

LEARNING GOALS	DESCRIPTION	WHEN MIGHT YOU USE THIS LESSON?
Students will be able to... • Break down a large project into smaller, bite-sized tasks.	Students read about how to break a project down into specific steps. Students then practice building their own Kanban boards to manage their progress on a project for class.	• Before students start a group project. • To encourage students to plan out a project before getting started.
CSTA STANDARDS	**COMMON CORE ELA STANDARDS**	**COMMON CORE MATH STANDARDS**
CSTA 2-AP-13: Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.	**CCRA.R.2:** Determine central ideas or themes of a text and analyze their development; summarize the key supporting details and ideas.	**MP1:** Make sense of problems and persevere in solving them.

LESSON 3: DESIGN (featuring Nicole Dominguez)

LEARNING GOALS	DESCRIPTION	WHEN MIGHT YOU USE THIS LESSON?
Students will be able to... • Define "user experience." • Explain how considering users' needs leads to better products.	Students read about how designers build user personas to improve their designs. Students then practice building their own user personas for a project they are currently working on in class, by considering the needs of their audience.	• Before students begin working on a paper or presentation. • To help students practice thinking about things from someone else's perspective.
CSTA STANDARDS	**COMMON CORE ELA STANDARDS**	**COMMON CORE MATH STANDARDS**
1B-AP-13: Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences.	**CCRA.W.4:** Produce clear and coherent writing in which the development, organization, and style are appropriate to task, purpose, and audience.	**N/A**

WHAT IS IT LIKE TO BE A COMPUTER SCIENTIST?

LESSON 4: ALGORITHMS (featuring Vanessa Tostado)

LEARNING GOALS	DESCRIPTION	WHEN MIGHT YOU USE THIS LESSON?
Students will be able to... • Define "algorithm." • Use flowcharts to design an algorithm to address a complex problem.	Students read about algorithms and flowcharts, then they practice drawing their own flowcharts to express an algorithm for a task they complete in their own lives.	• After students learn a new skill or process that you want them to be able to generalize (for example, writing a informative text or factoring an expression).
CSTA STANDARDS	**COMMON CORE ELA STANDARDS**	**COMMON CORE MATH STANDARDS**
2-AP-10: Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.	**CCRA.R.2:** Determine central ideas or themes of a text and analyze their development; summarize the key supporting details and ideas.	**MP8:** Look for and express regularity in repeated reasoning.

LESSON 5: THE DESIGN-BUILD-TEST CYCLE (featuring Miral Kotb)

LEARNING GOALS	DESCRIPTION	WHEN MIGHT YOU USE THIS LESSON?
Students will be able to... • Explain what happens in each stage of the design-build-test cycle. • Use the design-build-test cycle to create a paper prototype.	Students read about the design-build-test cycle, then they use it to create a paper prototype for their own mobile app idea.	• Before students start working on • a project. • To help students practice seeing feedback as an opportunity for innovation.
CSTA STANDARDS	**COMMON CORE ELA STANDARDS**	**COMMON CORE MATH STANDARDS**
2A-AP-15: Seek and incorporate feedback from team members and users to refine a solution that meets user needs.	**CCRA.SL.1:** Prepare for and participate effectively in a range of conversations and collaborations with diverse partners, building on others' ideas and expressing their own clearly and persuasively.	**MP3:** Construct viable arguments and critique the reasoning of others.

LESSON GOALS AND STANDARDS CROSSWALK

WHAT IS IT LIKE TO BE A COMPUTER SCIENTIST?

LESSON 6: DEBUGGING (featuring Grace Hopper)

LEARNING GOALS	DESCRIPTION	WHEN MIGHT YOU USE THIS LESSON?
Students will be able to... • Define debugging. • Debug a set of instructions. • Name at least two types of bugs.	Students learn about different types of bugs, practice finding and fixing bugs, and identify strategies for debugging.	• When you want students to practice close reading and identifying errors. • When you want students to evaluate sequences of events.
CSTA STANDARDS	**COMMON CORE ELA STANDARDS**	**COMMON CORE MATH STANDARDS**
1B-AP-15: Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.	**CCSS.ELA-LITERACY.CCRA.R.5:** Analyze the structure of texts, including how specific sentences, paragraphs, and larger portions of the text (e.g., a section, chapter, scene, or stanza) relate to each other and the whole.	**CCSS.MATH.PRACTICE.MP1:** Make sense of problems and persevere in solving them.

HOW IS COMPUTER SCIENCE CHANGING THE WORLD?

LESSON 7: HARDWARE AND SOFTWARE (featuring Katherine Johnson)

LEARNING GOALS	DESCRIPTION	WHEN MIGHT YOU USE THIS LESSON?
Students will be able to... • Explain the difference between hardware and software. • Model how hardware and software interact with each other.	Students learn the difference between hardware and software in abstract and practical terms and understand how they interact with one another through an interactive game.	• When you want students to identify tools to solve a specific problem. • When you want students to practice collaborating kinesthetically.
CSTA STANDARDS	**COMMON CORE ELA STANDARDS**	**COMMON CORE MATH STANDARDS**
1B-CS-02: Model how computer hardware and software work together as a system to accomplish tasks.	**CCSS.ELA-LITERACY.CCRA.SL.4:** Present information, findings, and supporting evidence such that listeners can follow the line of reasoning and the organization, development, and style are appropriate to task, purpose, and audience.	**CCSS.MATH.PRACTICE.MP6:** Attend to precision.

WHAT IS IT LIKE TO BE A COMPUTER SCIENTIST?

LESSON 8: PATTERN RECOGNITION (featuring the ENIAC Women)

LEARNING GOALS	DESCRIPTION	WHEN MIGHT YOU USE THIS LESSON?
Students will be able to… • Define pattern recognition. • Use pattern recognition to decode text with a cipher. • Name at least one way a computer can use pattern recognition.	Students learn about pattern recognition, practice encoding and decoding text using ciphers, and discuss how a computer can use pattern recognition.	• When you want students to practice translation skills. • When you want students to collaborate by problem-solving.
CSTA STANDARDS	**COMMON CORE ELA STANDARDS**	**COMMON CORE MATH STANDARDS**
2-NI-06: Apply multiple methods of encryption to model the secure transmission of information.	**CCSS.ELA-LITERACY.CCRA.R.1:** Read closely to determine what the text says explicitly and to make logical inferences from it; cite specific textual evidence when writing or speaking to support conclusions drawn from the text.	**CCSS.MATH.PRACTICE.MP2:** Reason abstractly and quantitatively.

HOW DO I ALREADY THINK LIKE A COMPUTER SCIENTIST?

LESSON 9: ABSTRACTION (featuring Brenda Wilkerson)

LEARNING GOALS	DESCRIPTION	WHEN MIGHT YOU USE THIS LESSON?
Students will be able to… • Define “abstraction.” • Perform abstraction. • Understand how abstraction applies in CS and elsewhere.	Students determine key features for various audiences, identify shared qualities of multiple objects, and visually represent abstracted information.	• When you want students to generalize information. • When you want students to compare and contrast.
CSTA STANDARDS	**COMMON CORE ELA STANDARDS**	**COMMON CORE MATH STANDARDS**
1A-AP-09: Model the way programs store and manipulate data by using numbers or other symbols to repre-sent information.	**CCRA.R.2:** Determine central ideas or themes of a text and analyze their development; summarize the key supporting details and ideas.	**CCSS.MATH.PRACTICE.MP7:** Look for and make use of structure.

HOW IS COMPUTER SCIENCE CHANGING THE WORLD?

LESSON 10: USER TESTING (featuring Chieko Asakawa)

LEARNING GOALS	DESCRIPTION	WHEN MIGHT YOU USE THIS LESSON?
Students will be able to... • Identify the needs of a specific user. • Modify a game to meet a user's needs. • Give specific, concrete feedback based on the needs of a user.	Students learn about user testing, design an activity for a particular user persona and practice giving and receiving feedback with their peers.	• When you want students to practice perspective-taking. • When you want students to practice giving and receiving feedback.
CSTA STANDARDS	**COMMON CORE ELA STANDARDS**	**COMMON CORE MATH STANDARDS**
1B-IC-19: Brainstorm ways to improve the accessibility and usability of technology products for the diverse needs and wants of users.	**CCSS.ELA-LITERACY.CCRA.SL.1:** Prepare for and participate effectively in a range of conversations and collaborations with diverse partners, building on others' ideas and expressing their own clearly and persuasively.	**CCSS.MATH.PRACTICE.MP5:** Use appropriate tools strategically.

ROBOTICS

HOW ARE ROBOTS CHANGING THE WORLD?

FEATURING: **AYANNA HOWARD**

Description: Students define the term robot, discuss real-life examples of robots, and have a discussion weighing the positive and negative impacts of robots on the world.

LEARNING GOALS

Students will be able to...
1. Define "robot."
2. Name 3 ways robots are used today.
3. Name 1 positive and 1 negative way robots are affecting the world.

MATERIALS

1. Worksheet
2. Video: What is a Robot? -WIRED
3. Photo: Ayanna Howard
4. Video: Ayanna Howard - MAKERS

STANDARDS

CSTA: 2-IC20
CCSS ELA: CCRA.R.2
CCSS Math: MP3

VOCABULARY

Robot: A machine that follows the sense-think-act pattern to carry out some physical action.

HOOK (10 MIN)

1. Set the stage by sharing the goals of the lesson.
2. Ask students to picture a robot. What does it look like? What does it do? Set a timer for 2 minutes while students write or draw their ideas.
3. Instruct students to turn to an elbow partner and share their ideas. As students share, circulate to see if there are any themes or misconceptions you'd like to discuss as a class.
4. As a class, have students share out the similarities and differences between what they and their partner thought. Summarize the discussion by sharing the fact that a robot can be challenging to define, partially because the field is changing so quickly!
5. Play the What is a Robot? Video.
   -Unplugged Alternative: Have students read the "What is a Robot?" section of the worksheet.
6. Highlight the definition for "robot" introduced in the video. Before moving on, take a poll of students by asking them to raise their hand if their original robot fit this definition.
7. Tell students that during the rest of the lesson, they're going to dig into real-life robots that living people have created! Challenge them to ask themselves: how are these robots impacting their world?

WOMAN IN TECH SPOTLIGHT (5 MIN)

1. Play the Ayanna Howard - MAKERS video.
   -Unplugged Alternative: Have students read the biography of Ayanna Howard included on the worksheet.
2. Explain: This video talked about two different kinds of robots that Ayanna worked on. One type was a robot that could navigate on Mars. The other type was for helping children with physical disabilities learn.
3. Ask: How are Ayanna's robots changing the world?
4. Suggested Answers: Ayanna's rovers are exploring space, which expands our understanding of the universe and how it works. Ayanna's educational tools are helping students of all ability levels have access to quality learning products.

ACTIVITY (25 MIN)

1. Split students into groups. Assign each group one of the following robots to research:
   - Self-driving cars: How Self-Driving Cars Work (Medium article)
   - Kuri: Mayfield Robotics' Kuri is an adorable home robot (video)
   - Chef bot: These robotic arms put a five-star chef in your kitchen (CNET video)
   - FarmBot: Meet FarmBot (video)
2. Set a timer for 20 minutes. Within their groups, students should fill out the worksheet based on their assigned robot. Then, groups should discuss:
   - How is your assigned robot changing the world?
   - In what ways are those changes positive or negative?
3. As a group: Share out the robot you discussed and its impact. The best student answers will also address the ways their robot senses, thinks, and acts.

DEBRIEF (5 MIN)

1. Explain to students that they're going to take a final vote on the impact of robots.
2. Designate one side of the room to be "helping" and one side to be "hurting."
3. Give students a moment to consider whether robots are helping or hurting the world.
4. Have students move themselves based on their opinions. Students might place themselves somewhere in the middle.
5. Have students turn to someone standing nearby, with a similar opinion. Set a timer for 1 minute while they discuss why they agree with each other.
6. Have students find a different partner who has a different opinion from them. Set a timer for 2 minutes while they discuss their different perspectives.

7. As a group, ask a few students to summarize their discussions with their partners. Student answers may vary, but some sample responses are included below:
   - Robots are helping the world because they can automate tasks that are hard or time-consuming for humans to complete manually. That frees up humans to do more creative work, which they might enjoy more.
   - Robots are potentially harmful because automating tasks means that people who used to do that work lose their jobs.
   - Robots are neither wholly good or wholly bad on their own. They can be either helpful or harmful, depending on who is building them and what they are being programmed to do.
8. Close out by sharing that robotics is a growing field that they can continue to study.

WHAT IS A ROBOT?

A robot is a machine that follows a sense-think-act pattern to carry out some physical action. This means that the robot is programmed to follow three different kinds of steps:

SENSE. First, the robot uses small electronic parts called sensors to take in information about the world around it. Some of these sensors are similar to your own senses of sight (like a camera) or hearing (like a microphone). But there are also other kinds of sensors that measure things like temperature, speed, or even fingerprints!

THINK. Next, the robot processes the information it sensed, so that it can decide what to do next. For example, imagine an assistant robot. First, it might sense through its microphone that you said something. Then it would run some code to break down what you said into specific commands. Then it would decide what action to perform, based on the command you gave.

ACT. Finally, the robot has to do some physical action. The physical movement is what separates robots from other "thinking" machines like laptops or cell phones. This action could be something as simple as moving forward or backward, or it could be something more complex, like moving a ball into a bucket. In addition to the physical action, the robot might use other types of output, like displaying some information displaying on a screen or playing an audio message through a speaker.

There are tons and tons of different kinds of robots in the world today. While some robots look like the metal human-like figures you might have seen in movies (think WALL-E), you have probably encountered different kinds of robots without even realizing it!

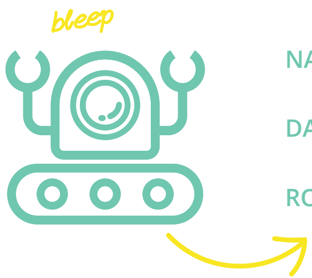
AYANNA HOWARD

Ayanna Howard is a roboticist and professor at the Georgia Institute of Technology. She first became interested in robotics after watching a TV show called The Bionic Woman, which is a sci-fi show about a woman with cybernetic implants that give her superhuman powers. Ayanna liked that the show told a story about how technology can be used for social good.

Ayanna earned a bachelor's degree in engineering, a master's degree in electrical engineering, a business degree, and a PhD in electrical engineering, robotics, and computer science! While she was in school, Ayanna started working at NASA's Jet Propulsion Laboratory, where she built rovers to navigate on Mars by themselves.

When Ayanna was working as a professor at Georgia Tech, she wanted to shift her focus from using robots in space to using robots for health care. It was a new area that not many other people had worked in before, which Ayanna thought was a little scary but also exciting! Ayanna founded Zyrobotics, a company that builds educational learning tools for children with physical limitations. One product they created was an electronic turtle stuffed animal named Zumo. Children could press on different parts of Zumo's shell, which would then map to different on-screen gestures (like swiping or tapping) for a mobile app.

How are Ayannas' robots changing the world?



NAME: ____________________

DATE: ____________________

ROBOT: ____________________

THINK	**WHAT DECISIONS DOES YOUR ROBOT MAKE?**
SENSE	**HOW DOES YOUR ROBOT TAKE IN INFORMATION?**
ACT	**WHAT PHYSICAL ACTION DOES YOUR ROBOT DO?**

DISCUSS THE FOLLOWING QUESTIONS WITH YOUR GROUP. RECORD NOTES FROM YOUR DISCUSSION HERE:

How is your assigned robot changing the world?

In what ways are those changes positive or negative?

LESSON GOALS AND STANDARDS CROSSWALK

HOW IS COMPUTER SCIENCE CHANGING THE WORLD?

LESSON 1: ROBOTICS (featuring Ayanna Howard)

LEARNING GOALS	DESCRIPTION	WHEN MIGHT YOU USE THIS LESSON?
Students will be able to... • Define the term "robot." • Name 3 ways robots are used today. • Name 1 positive and 1 negative way robots are affecting the world.	Students learn what a robot is, read and talk about real-world examples of different kinds of robots, and have a discussion weighing the positive and negative impacts of robots on the world.	• To provide practice reading and discussing nonfiction texts. • To provide practice deconstructing and evaluating the arguments of others.
CSTA STANDARDS	**COMMON CORE ELA STANDARDS**	**COMMON CORE MATH STANDARDS**
2-IC-20: Compare tradeoffs associated with computing technologies that affect people's everyday activities and career options.	**CCRA.R.2:** Determine central ideas or themes of a text and analyze their development; summarize the key supporting details and ideas.	**MP3:** Construct viable arguments and critique the reasoning of others.

HOW DO I ALREADY THINK LIKE A COMPUTER SCIENTIST?

LESSON 2: DECOMPOSITION (featuring Ada Lovelace)

LEARNING GOALS	DESCRIPTION	WHEN MIGHT YOU USE THIS LESSON?
Students will be able to... • Break down a large project into smaller, bite-sized tasks.	Students read about how to break a project down into specific steps. Students then practice building their own Kanban boards to manage their progress on a project for class.	• Before students start a group project. • To encourage students to plan out a project before getting started.
CSTA STANDARDS	**COMMON CORE ELA STANDARDS**	**COMMON CORE MATH STANDARDS**
CSTA 2-AP-13: Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.	**CCRA.R.2:** Determine central ideas or themes of a text and analyze their development; summarize the key supporting details and ideas.	**MP1:** Make sense of problems and persevere in solving them.

LESSON 3: DESIGN (featuring Nicole Dominguez)

LEARNING GOALS	DESCRIPTION	WHEN MIGHT YOU USE THIS LESSON?
Students will be able to... • Define "user experience." • Explain how considering users' needs leads to better products.	Students read about how designers build user personas to improve their designs. Students then practice building their own user personas for a project they are currently working on in class, by considering the needs of their audience.	• Before students begin working on a paper or presentation. • To help students practice thinking about things from someone else's perspective.
CSTA STANDARDS	**COMMON CORE ELA STANDARDS**	**COMMON CORE MATH STANDARDS**
1B-AP-13: Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences.	**CCRA.W.4:** Produce clear and coherent writing in which the development, organization, and style are appropriate to task, purpose, and audience.	**N/A**

WHAT IS IT LIKE TO BE A COMPUTER SCIENTIST?

LESSON 4: ALGORITHMS (featuring Vanessa Tostado)

LEARNING GOALS	DESCRIPTION	WHEN MIGHT YOU USE THIS LESSON?
Students will be able to... • Define "algorithm." • Use flowcharts to design an algorithm to address a complex problem.	Students read about algorithms and flowcharts, then they practice drawing their own flowcharts to express an algorithm for a task they complete in their own lives.	• After students learn a new skill or process that you want them to be able to generalize (for example, writing a informative text or factoring an expression).
CSTA STANDARDS	**COMMON CORE ELA STANDARDS**	**COMMON CORE MATH STANDARDS**
2-AP-10: Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.	**CCRA.R.2:** Determine central ideas or themes of a text and analyze their development; summarize the key supporting details and ideas.	**MP8:** Look for and express regularity in repeated reasoning.

LESSON 5: THE DESIGN-BUILD-TEST CYCLE (featuring Miral Kotb)

LEARNING GOALS	DESCRIPTION	WHEN MIGHT YOU USE THIS LESSON?
Students will be able to… • Explain what happens in each stage of the design-build-test cycle. • Use the design-build-test cycle to create a paper prototype.	Students read about the design-build-test cycle, then they use it to create a paper prototype for their own mobile app idea.	• Before students start working on • a project. • To help students practice seeing feedback as an opportunity for innovation.
CSTA STANDARDS	**COMMON CORE ELA STANDARDS**	**COMMON CORE MATH STANDARDS**
2A-AP-15: Seek and incorporate feedback from team members and users to refine a solution that meets user needs.	**CCRA.SL.1:** Prepare for and participate effectively in a range of conversations and collaborations with diverse partners, building on others' ideas and expressing their own clearly and persuasively.	**MP3:** Construct viable arguments and critique the reasoning of others.

DECOMPOSITION

HOW DO YOU BREAK DOWN PROBLEMS LIKE A COMPUTER SCIENTIST?

FEATURING: **ADA LOVELACE**

Description: Students read about how to break a project down into specific steps. Students then practice building their own Kanban boards to manage their progress on a project for class.

LEARNING GOALS

Students will be able to...

1. Break down a large project into smaller, bite-sized tasks.

MATERIALS

1. Worksheet
2. Photo: Ada Lovelace
3. Poster paper and sticky notes (if students are creating a physical Kanban board)

STANDARDS

CSTA: 2-AP-13
CCSS ELA: CCRA.R.2
CCSS Math: MP1

VOCABULARY

Decomposition: The act of breaking something down into smaller pieces.
Kanban board: A tool used for keeping track of progress on a project. The board has three columns labeled "To Do," "Doing," and "Done."

WOMAN IN TECH SPOTLIGHT (5 MIN)

1. Ask students to close their eyes and imagine the very first computer programmer. What do they look like? What does their work look like? What era is it?
2. Distribute the worksheet to students. Set a timer for 3 minutes while students read the spotlight on Ada Lovelace.
3. Have students share out how the bio about Ada Lovelace compared to what they imagined about the first computer programmer.

ACTIVITY (35 MIN)

1. Explain: Today you're going to learn about decomposition. Decomposition is the process of breaking down something complex into smaller parts. This is the same process Ada used when she wrote an algorithm for how to compute a sequence of numbers. Computer scientists use decomposition to build algorithms, but they also use it to plan out projects. Today, we're going to focus on this second case.
2. Set a timer for 3 minutes while students read the "What is Decomposition?" section of the worksheet.

3. Set a timer for 5 minutes while students fill out the table on the worksheet by improving the poorly written tasks. Have students share out the improved tasks they wrote. Some suggested answers are included:

Write report	Too large a task	Find 3 sources about global warming. Choose 1 quote from each source. Add quotes from sources into report outline. Write introduction paragraph. Write 1 supporting paragraph for each source. Write conclusion paragraph.
Email Keya	Hard to understand	Email Keya the links to the articles for the bibliography.
Poster	Not specific	Rewrite lab report content for poster. Print out content for poster. Assemble poster. Decorate poster.

4. Introduce the project you want students to start planning for. This could be any kind of large project (a presentation, a report, a diorama, etc.) that requires students to complete multiple steps.
5. Set a timer for 5 minutes while students break the assigned project down into a list of tasks. Students should record their work in the second table on the worksheet. Circulate while students work and redirect them if you notice their tasks are not specific, small, or understandable.
   -If students are working in groups, they may require more time.
6. Explain: Now that you have a list of what you need to get done for this project, the next step is to turn that list into a tool that can help you keep track of your progress.
7. Set a timer for 3 minutes while students read the "What is Kanban?" section of the worksheet.
8. Set a timer for 10 minutes while students build their own Kanban boards. If you have space, students can use poster paper to create their own physical boards. Alternatively, students can use an online tool like Trello to create a digital board.
   -If you choose to have students create Trello boards, be sure to have them invite you to the board so you can see their progress.

DEBRIEF (5 MIN)

1. Set a timer for 2 minutes. Have students partner up and share their task lists and Kanban boards. Then have students discuss the following with their partner:
   -What was easy about this activity? What was challenging?
   -When else might you use decomposition in your daily life?
2. Ask for a few students to share out their responses. The best student answers will emphasize that decomposition can be used any time a complex problem has to be broken down into smaller chunks. For example, decomposition can be used to solve a multi-step math problem or for planning a surprise party.
3. Explain to students that if decomposition was interesting to them, they might consider studying it further by learning more about computer science online or in classes!

WHAT IS DECOMPOSITION?

Before you can get started on a new project, you first need to figure out everything that needs to get done! One way to do that is by breaking down the big project into smaller parts, like making a to-do list. This process is called **decomposition.** Once you have a list of all the tasks you need to complete for your project, you can figure out the best path forward.

There are different ways you might choose to break down your project. You might choose to separate tasks by project phase. For example, if you're writing an essay, you might have some tasks for a draft phase, a writing phase, and an editing phase. Or you might choose to split tasks by category. For example, if you're making a science fair project, you might have some tasks for your experiment, some for a lab report, and some for a poster. But no matter how you choose to group things, the most important thing is to make sure the tasks themselves are written well.

A good task will be:

-**Specific.** It should be clear exactly what needs to be done. For example, "Find 3 articles about the positive and negative effects of social media" is better than "social media articles."
-**Something you can finish in one sitting.** If it seems like it will take you more than an hour to complete, try breaking it down into multiple steps.
-**Easy to understand.** Make sure you and your teammates will be able to remember what you meant! Using specific words will help with this.

ADA LOVELACE

Ada Lovelace was born in 1815 in London. Although it was highly unusual for the time, Ada's mother insisted that Ada have tutors in math and science. When Ada was 17, she met mathematician Charles Babbage, a professor at the University of Cambridge. Charles was impressed by Ada's mathematical mind and called her an "enchantress of numbers." Ada and Charles remained lifelong friends.

In 1837, Charles had an idea for a new kind of machine called the "analytical engine." Although it was never built - it is considered to be the ancestor of the modern computers used today. Ada saw the potential of the idea, and in 1843 she translated a description of the engine from French to English. As she translated, she corrected mistakes that Charles had originally made and added her own notes to the translation - they were three times longer than the original! Her notes included the first published computer algorithm: it broke down how the machine could be used to compute a complex sequence of numbers. Because of this algorithm, Ada Lovelace is often called the world's first programmer.

NAME: ______________________

DATE: ______________________

INSTRUCTIONS: Use the table below to practice writing good tasks. For each task, figure out what's wrong with it, and then write an improved version of the same task. In some cases, you might need to make up some details or break the given task into multiple smaller tasks.

POORLY WRITTEN TASK	WHAT'S WRONG WITH IT?	BETTER TASK(S)
Write report		
Email Kiya		
Write report		

INSTRUCTIONS: Think of a project you are working on. Use the space below to break down your project into tasks.

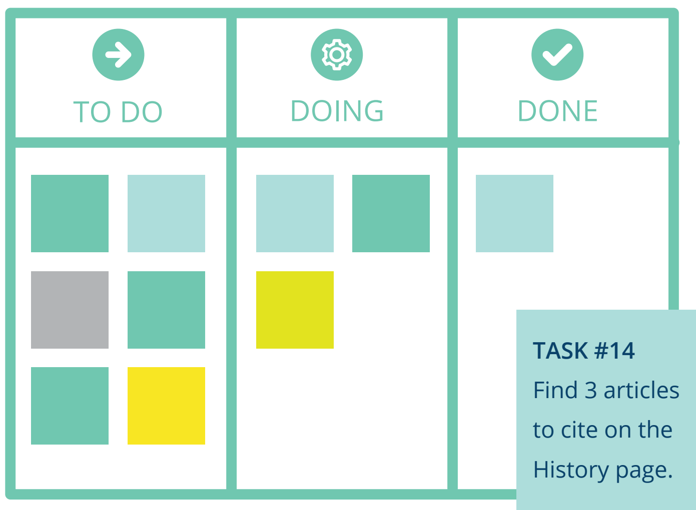
PROJECT

NAME: ______________________________

DATE: ______________________________

WHAT IS A KANBAN BOARD?

Once a team has broken down their project into tasks, they can use the Kanban method - a strategy for keeping track of progress on a project. In the Kanban method, all of the tasks for a project are displayed on a large board so that everyone can see what needs to be done. Then, tasks are organized by placing the most urgent tasks at the top of the list. The board is divided into 3 or more columns, labeled by level of completion: "To Do," "Doing," and "Done." Kanban is useful because everyone on the team can see the work that is being done in one place.



To use Kanban, do the following:

1. Write each task on its own card and place it in the "To Do" column.

2. Once a person takes responsibility for a task, write their name on it and move it to the "Doing" column.

When the task is complete, move it to the "Done" column.

INSTRUCTIONS:

3. Write each task for your project on its own index card or sticky note.

4. Build a Kanban board for your team. Add all your tasks to your board.

5. Use your Kanban board to plan a timeline for your project. How might you spread out the different milestones or checkpoints for your project between now and when your project is due?

6. When else might you use decomposition in your daily life?

LESSON GOALS AND STANDARDS CROSSWALK

HOW IS COMPUTER SCIENCE CHANGING THE WORLD?

LESSON 1: ROBOTICS (featuring Ayanna Howard)

LEARNING GOALS	DESCRIPTION	WHEN MIGHT YOU USE THIS LESSON?
Students will be able to... • Define the term "robot." • Name 3 ways robots are used today. • Name 1 positive and 1 negative way robots are affecting the world.	Students learn what a robot is, read and talk about real-world examples of different kinds of robots, and have a discussion weighing the positive and negative impacts of robots on the world.	• To provide practice reading and discussing nonfiction texts. • To provide practice deconstructing and evaluating the arguments of others.
CSTA STANDARDS	**COMMON CORE ELA STANDARDS**	**COMMON CORE MATH STANDARDS**
2-IC-20: Compare tradeoffs associated with computing technologies that affect people's everyday activities and career options.	**CCRA.R.2:** Determine central ideas or themes of a text and analyze their development; summarize the key supporting details and ideas.	**MP3:** Construct viable arguments and critique the reasoning of others.

HOW DO I ALREADY THINK LIKE A COMPUTER SCIENTIST?

LESSON 2: DECOMPOSITION (featuring Ada Lovelace)

LEARNING GOALS	DESCRIPTION	WHEN MIGHT YOU USE THIS LESSON?
Students will be able to... • Break down a large project into smaller, bite-sized tasks.	Students read about how to break a project down into specific steps. Students then practice building their own Kanban boards to manage their progress on a project for class.	• Before students start a group project. • To encourage students to plan out a project before getting started.
CSTA STANDARDS	**COMMON CORE ELA STANDARDS**	**COMMON CORE MATH STANDARDS**
CSTA 2-AP-13: Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.	**CCRA.R.2:** Determine central ideas or themes of a text and analyze their development; summarize the key supporting details and ideas.	**MP1:** Make sense of problems and persevere in solving them.

LESSON 3: DESIGN (featuring Nicole Dominguez)

LEARNING GOALS	DESCRIPTION	WHEN MIGHT YOU USE THIS LESSON?
Students will be able to... • Define "user experience." • Explain how considering users' needs leads to better products.	Students read about how designers build user personas to improve their designs. Students then practice building their own user personas for a project they are currently working on in class, by considering the needs of their audience.	• Before students begin working on a paper or presentation. • To help students practice thinking about things from someone else's perspective.
CSTA STANDARDS	**COMMON CORE ELA STANDARDS**	**COMMON CORE MATH STANDARDS**
1B-AP-13: Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences.	**CCRA.W.4:** Produce clear and coherent writing in which the development, organization, and style are appropriate to task, purpose, and audience.	**N/A**

WHAT IS IT LIKE TO BE A COMPUTER SCIENTIST?

LESSON 4: ALGORITHMS (featuring Vanessa Tostado)

LEARNING GOALS	DESCRIPTION	WHEN MIGHT YOU USE THIS LESSON?
Students will be able to... • Define "algorithm." • Use flowcharts to design an algorithm to address a complex problem.	Students read about algorithms and flowcharts, then they practice drawing their own flowcharts to express an algorithm for a task they complete in their own lives.	• After students learn a new skill or process that you want them to be able to generalize (for example, writing a informative text or factoring an expression).
CSTA STANDARDS	**COMMON CORE ELA STANDARDS**	**COMMON CORE MATH STANDARDS**
2-AP-10: Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.	**CCRA.R.2:** Determine central ideas or themes of a text and analyze their development; summarize the key supporting details and ideas.	**MP8:** Look for and express regularity in repeated reasoning.

LESSON 5: THE DESIGN-BUILD-TEST CYCLE (featuring Miral Kotb)

LEARNING GOALS	DESCRIPTION	WHEN MIGHT YOU USE THIS LESSON?
Students will be able to... • Explain what happens in each stage of the design-build-test cycle. • Use the design-build-test cycle to create a paper prototype.	Students read about the design-build-test cycle, then they use it to create a paper prototype for their own mobile app idea.	• Before students start working on • a project. • To help students practice seeing feedback as an opportunity for innovation.
CSTA STANDARDS	**COMMON CORE ELA STANDARDS**	**COMMON CORE MATH STANDARDS**
2A-AP-15: Seek and incorporate feedback from team members and users to refine a solution that meets user needs.	**CCRA.SL.1:** Prepare for and participate effectively in a range of conversations and collaborations with diverse partners, building on others' ideas and expressing their own clearly and persuasively.	**MP3:** Construct viable arguments and critique the reasoning of others.

USER EXPERIENCE

HOW DO YOU BREAK DOWN PROBLEMS LIKE A COMPUTER SCIENTIST?

FEATURING: **NICOLE DOMINGUEZ**

Description: Students read about how designers think about their users. Students then practice building their own user personas for a project they are currently working on in class, by considering the needs of their audience.

LEARNING GOALS

Students will be able to...

1. Define "user experience."
2. Explain how considering users' needs leads to better products.

MATERIALS

1. Worksheet
2. Photo: Nicole Dominguez

STANDARDS

CSTA: 1B-AP-13
CCSS ELA: CCRA.W.4

VOCABULARY

User experience: The feelings and attitudes a person has when they are using a product.
User persona: A made-up profile for a person who matches the characteristics of someone who might use a particular product

HOOK (5 MIN)

1. Ask students to think about a time when they had a really bad experience using a website, app, or other product.
   -Depending on your classroom content, you might choose to have students think about an experience with a book, paper, or other relevant product.
2. Set a timer for 2 minutes. Have students turn to a partner and share the following:
   -What was the product?
   -What were you trying to accomplish?
   -Why was it a frustrating experience?
3. Explain: As you may already know from personal experience, the best products are built when they take the user's experience into account. This means thinking about things like what the user wants from your product and how they might interact with it. User experience is something the best designers think about from the very beginning of a project.

WOMAN IN TECH SPOTLIGHT (5 MIN)

1. Set a timer for 3 minutes while students read the spotlight on Nicole Dominguez included on the worksheet.
2. Have students turn to a partner and discuss why Nicole might think about user experience.
   -Suggested Answer: By thinking about user experience, Nicole can build better products that fit what her clients want. That will make her clients happy, which means they might recommend her for other projects in the future.

ACTIVITY (30 MIN)

1. Set a timer for 5 minutes while students read the section of the worksheet titled "What is User Experience?"
2. Set a timer for 2 minutes. Have students turn to a partner and discuss the following questions:
   -What is a user persona?
   -How are user personas related to user experience?
3. Ask a few students to share out and summarize what they and their partner discussed.
4. Ask students to think about a project they are currently working on. This could be a program, an essay, a research project, or any other class assignment with an intended audience.
5. Explain to students that they will now have a chance to practice building their own user personas for that project.
6. Set a timer for 15 minutes while students fill out the user personas on the worksheet. Encourage them to be as detailed and creative as they'd like. The goal is for them to think about what different kinds of people might need from their project.

DEBRIEF (5 MIN)

1. Set a timer for 2 minutes. Have students turn to a new partner and share the user personas they created. Then have students discuss the following with their partner: How does thinking about your audience like a computer scientist help you create a better project?
2. Ask for a few students to share out their responses. The best student answers will emphasize that building user personas and thinking about what other people need from your project lets you build something more useful, which ultimately makes your project more successful.
3. As you close out the class, tell students that they will use the user personas they wrote as they continue to develop their project - just like computer scientists might as they build new products!

WHAT IS USER EXPERIENCE?

User experience is the way a person feels when they are using a product. When a product has a good user experience, that usually means that a person finds it enjoyable and easy to use. But a good user experience isn't something that just gets built into a product automatically. The best designers think about how to build a great user experience from the very beginning of a project.

One way that designers make sure their product has a good user experience is by thinking like an actual user! Before they even start drawing designs, designers will often hold user interviews to ask real humans about their experiences. The designers might ask about the users' backgrounds or about what kinds of things they want out of a product. After talking to lots of different people, the designers look for trends and make categories of the different kinds of users who are interested in their product. Then, they use those categories to write user personas.

A user persona is like a profile page for a made-up person who matches some characteristics of someone who might use your product. The actual details in the persona vary based on what kinds of details are important to your particular project. It might include a short bio of that person's life, their age, their job, or their level of comfort with different technology. User personas help designers make sure they are designing a product that fits the needs of a specific type of user. For example, it can be hard to think about what kind of design might be fun for children ages 8-12, but it's easier to think about what might be fun for your 10-year-old cousin, Nina. Two different user personas for a mobile app to help students apply to colleges are pictured below:

NAME: Carla
AGE:17

BIO: Carla is a junior in high school. When she's not at basketball practice, she can be found in the computer lab, building cool websites with her friends. She's interested in exploring colleges that offfer computer science majors and scholarships for student athletes.

FAVORITE APPS: Snapschat, ESPN, Spotify

NAME: Faatimah
AGE:18

BIO:Faatimah is a senior in high school and is currently applying to colleges. She wants to double major in physics and art history. Her biggest concern is keeping track of all her applications: deadlines, essays, and letters of recommendation.

FAVORITE APPS: Instagram, Angry Bird, Twitter

NICOLE DOMINGUEZ

Nicole Dominguez learned to program when she was 13 by building her own blog about her life and the things she was creating. She started freelancing when she was 15, working with some friends to take on client projects. Now, she is a user interface (UI) designer and front-end developer. That means she builds the parts of websites that users interact with directly. She likes working at the intersection between design and code.

Nicole used to be a digital nomad, which means she worked remotely (doing all her work online from her computer, instead of in a normal office setting) and used the money she earned to travel the world. It gave her space to think about her own goals and what she wants out of life.

NAME: ______________________

DATE: ______________________

INSTRUCTIONS: Think about a project you are currently working on. Who is your target audience? What kinds of people might they be? Use the table below to create two different user personas for two types of people who might interact with your project.

PROJECT

NAME: AGE:	BIO: WHAT THEY NEED FROM YOUR PROJECT: ADDITIONAL DETAILS:

NAME: AGE:	BIO: WHAT THEY NEED FROM YOUR PROJECT: ADDITIONAL DETAILS:

How does thinking about your audience helps you create a better project?

LESSON GOALS AND STANDARDS CROSSWALK

HOW IS COMPUTER SCIENCE CHANGING THE WORLD?

LESSON 1: ROBOTICS (featuring Ayanna Howard)

LEARNING GOALS	DESCRIPTION	WHEN MIGHT YOU USE THIS LESSON?
Students will be able to... • Define the term "robot." • Name 3 ways robots are used today. • Name 1 positive and 1 negative way robots are affecting the world.	Students learn what a robot is, read and talk about real-world examples of different kinds of robots, and have a discussion weighing the positive and negative impacts of robots on the world.	• To provide practice reading and discussing nonfiction texts. • To provide practice deconstructing and evaluating the arguments of others.
CSTA STANDARDS	**COMMON CORE ELA STANDARDS**	**COMMON CORE MATH STANDARDS**
2-IC-20: Compare tradeoffs associated with computing technologies that affect people's everyday activities and career options.	**CCRA.R.2:** Determine central ideas or themes of a text and analyze their development; summarize the key supporting details and ideas.	**MP3:** Construct viable arguments and critique the reasoning of others.

HOW DO I ALREADY THINK LIKE A COMPUTER SCIENTIST?

LESSON 2: DECOMPOSITION (featuring Ada Lovelace)

LEARNING GOALS	DESCRIPTION	WHEN MIGHT YOU USE THIS LESSON?
Students will be able to... • Break down a large project into smaller, bite-sized tasks.	Students read about how to break a project down into specific steps. Students then practice building their own Kanban boards to manage their progress on a project for class.	• Before students start a group project. • To encourage students to plan out a project before getting started.
CSTA STANDARDS	**COMMON CORE ELA STANDARDS**	**COMMON CORE MATH STANDARDS**
CSTA 2-AP-13: Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.	**CCRA.R.2:** Determine central ideas or themes of a text and analyze their development; summarize the key supporting details and ideas.	**MP1:** Make sense of problems and persevere in solving them.

LESSON 3: DESIGN (featuring Nicole Dominguez)

LEARNING GOALS	DESCRIPTION	WHEN MIGHT YOU USE THIS LESSON?
Students will be able to... • Define "user experience." • Explain how considering users' needs leads to better products.	Students read about how designers build user personas to improve their designs. Students then practice building their own user personas for a project they are currently working on in class, by considering the needs of their audience.	• Before students begin working on a paper or presentation. • To help students practice thinking about things from someone else's perspective.
CSTA STANDARDS	**COMMON CORE ELA STANDARDS**	**COMMON CORE MATH STANDARDS**
1B-AP-13: Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences.	**CCRA.W.4:** Produce clear and coherent writing in which the development, organization, and style are appropriate to task, purpose, and audience.	**N/A**

WHAT IS IT LIKE TO BE A COMPUTER SCIENTIST?

LESSON 4: ALGORITHMS (featuring Vanessa Tostado)

LEARNING GOALS	DESCRIPTION	WHEN MIGHT YOU USE THIS LESSON?
Students will be able to... • Define "algorithm." • Use flowcharts to design an algorithm to address a complex problem.	Students read about algorithms and flowcharts, then they practice drawing their own flowcharts to express an algorithm for a task they complete in their own lives.	• After students learn a new skill or process that you want them to be able to generalize (for example, writing a informative text or factoring an expression).
CSTA STANDARDS	**COMMON CORE ELA STANDARDS**	**COMMON CORE MATH STANDARDS**
2-AP-10: Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.	**CCRA.R.2:** Determine central ideas or themes of a text and analyze their development; summarize the key supporting details and ideas.	**MP8:** Look for and express regularity in repeated reasoning.

LESSON 5: THE DESIGN-BUILD-TEST CYCLE (featuring Miral Kotb)

LEARNING GOALS	DESCRIPTION	WHEN MIGHT YOU USE THIS LESSON?
Students will be able to… • Explain what happens in each stage of the design-build-test cycle. • Use the design-build-test cycle to create a paper prototype.	Students read about the design-build-test cycle, then they use it to create a paper prototype for their own mobile app idea.	• Before students start working on • a project. • To help students practice seeing feedback as an opportunity for innovation.
CSTA STANDARDS	**COMMON CORE ELA STANDARDS**	**COMMON CORE MATH STANDARDS**
2A-AP-15: Seek and incorporate feedback from team members and users to refine a solution that meets user needs.	**CCRA.SL.1:** Prepare for and participate effectively in a range of conversations and collaborations with diverse partners, building on others' ideas and expressing their own clearly and persuasively.	**MP3:** Construct viable arguments and critique the reasoning of others.

ALGORITHMS

WHAT IS IT LIKE TO BUILD ALGORITHMS LIKE A COMPUTER SCIENTIST?

FEATURING: **VANESSA TOSTADO**

Description: Students read about algorithms and flowcharts, then they practice drawing their own flowcharts to express an algorithm for a task they complete in their own lives.

LEARNING GOALS

Students will be able to...

1. Define "algorithm."
2. Use flowcharts to design an algorithm to address a complex problem.

MATERIALS

1. Worksheet
2. Photo: Vanessa Tostado
3. Additional blank paper and drawing materials for flowcharts

STANDARDS

CSTA: 2-AP-10
CCSS ELA: CCRA.R.2
CCSS Math: MP8

VOCABULARY

Algorithm: A list of step-by-step instructions for how to complete a task.
Flowchart: A diagram that uses different symbols and arrows to show the different steps in a process.

HOOK (10 MIN)

1. Ask: What are some chores that you do on a regular basis? Have students call out some tasks. For example: "Do the dishes." "Walk my dog." "Take out the trash."
2. Ask students to imagine that they could program a robot to do those chores instead. Explain that, in order to build that robot, students would need to know how to design algorithms that tell the robot what to do. However, robots aren't very smart, so this will be a challenge in planning out your ideas and writing instructions with precision.
3. In the rest of this activity, students will practice designing their own algorithms using a tool called a flowchart.
4. Distribute worksheets to students, and set a timer for 5 minutes while students read the first two sections: "What is an Algorithm?" and "Designing an Algorithm: Flowcharts."
5. As a group, have a few students summarize what an algorithm is, what a flowchart is, and what the different symbols in a flowchart represent.
6. Choose one of the chores that students called out at the beginning of the activity. The best examples will include one or more decision points, to make the algorithm slightly more complex.
7. As a group, have students help you draw a flowchart on the board for how to complete that chore.

WOMAN IN TECH SPOTLIGHT (5 MIN)

1. Set a timer for 2 minutes while students read the spotlight on Vanessa Tostado.
2. Set a timer for 2 minutes while students work with an elbow partner to sketch a flowchart for an algorithm of how Vanessa's Tag It! app might work.
3. Have students group up with another pair and share their flowcharts.

ACTIVITY (25 MIN)

1. Have students choose a more complex task to design an algorithm for.
2. You might choose to have students focus on a topic related to your specific class content (e.g., how to solve a system of equations, how to write a persuasive essay, how to find sources for a research project), or you may let students choose a task for themselves.
- Ideally, this task should be something that can be completed in a few minutes, so that students can have a chance to practice following each other's algorithms.

3. Set a timer for 5 minutes while students work with a partner to fill out the top three rows of the table on the worksheet, which cover basic information for their algorithm, like inputs, outputs, steps, and decision points.
4. Set a timer for 10 minutes while students design and sketch their flowcharts.
– Encourage students to test out their algorithms using a variety of inputs. For example, if students are writing algorithms on how to proofread a paper, they should try applying their algorithm to a variety of different examples of student writing.

5. Have students trade flowcharts with another group and take turns trying to follow each other's algorithms.
- Tell students to do exactly what the instructions say, not what they think the instructions meant to say. That might mean that the algorithm ends up not working, but that's okay! This is how we learn! All good computer scientists have to go through many rounds of testing before they have a finished product.
- Students should give each other feedback on what parts of the algorithm are clear and what parts need more revision.

DEBRIEF (5 MIN)

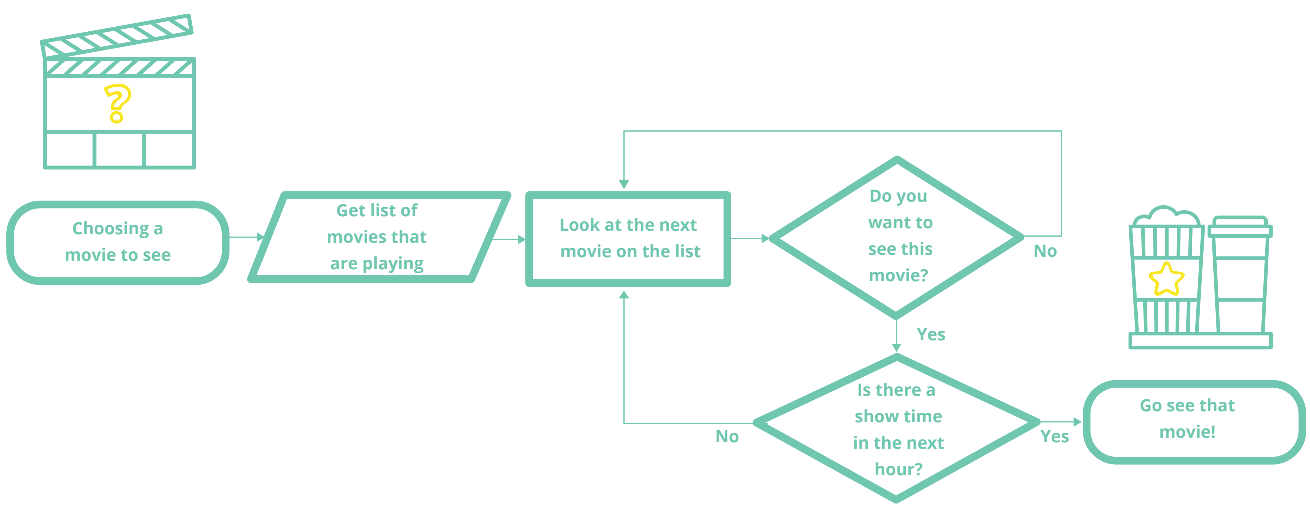
1. Facilitate a group discussion using the following questions:
- What was interesting or challenging about trying to build your algorithm? What strategies did you use to build your flowchart?
- What did you learn from having someone else try to act out your algorithm? Is there anything about your flowchart that you'd revise, after seeing someone try to use it?
- How is designing instructions for a computer different from giving instructions to another human being?
    - Suggested Answer: Instructions for a computer have to be very specific. A human can make inferences and understand what you were trying to say, but a computer will only do exactly what you tell it to do, even if that means it does the wrong thing.

WHAT IS AN ALGORITHM?

Computers aren't smart on their own. They require humans called programmers to write the instructions (also called code) that tell them how to do things. But how do programmers know what instructions to write? First, they start by designing an **algorithm**. An algorithm is a set of step-by-step instructions on how to complete a particular task. You can think of an algorithm like a recipe; it has some inputs (the ingredients) which, if you follow the instructions correctly, get turned into an output (the finished dish).

DESIGNING AN ALGORITHM: FLOWCHARTS

One strategy that computer scientists use to design algorithms is drawing flowcharts. A flowchart is a diagram that uses different kinds of symbols and arrows to show the different steps in a process. For example, the flowchart below shows one possible algorithm for how to decide what movie to go see.



The different kinds of symbols each have a different meaning:

- Ovals show the start and end of a process.
- Parallelograms show inputs or outputs for the process.
- Rectangles show steps or smaller processes that make up the larger task.
- Diamonds show decision points, also called branches.
- Arrows show connections and relationships between the other parts of the process.

When Vanessa Tostado was growing up in East Palo Alto, California, she and her friends noticed that there was a lot of trash and graffiti around their neighborhood. The team of girls worked together to build Tag It!, a mobile app that lets people from the community mark locations that need to be cleaned up on a map. Other people in the community can then see the map and go remove the graffitti or pick up the trash. They entered their project in the Technovation Challenge, and they came in 5th place in the Bay Area and 20th place worldwide!

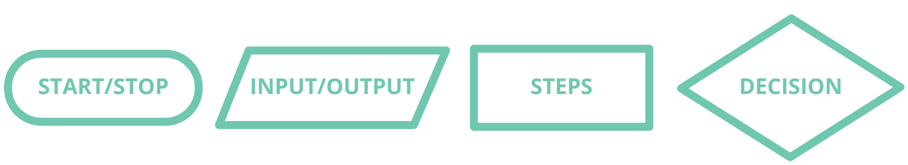
Vanessa is currently a student at Wesleyan University in Connecticut, where she is majoring in computer science. She has spent summers interning at companies like LinkedIn, Stanford Children's Health, and the Environmental Defense Fund.

NAME: ______________________

DATE: ______________________

INSTRUCTIONS: Choose a task or process that you do on a regular basis. Use the table below to create a flowchart that shows an algorithm for how someone might complete that task.

TASK:	
INPUTS:	**OUTPUTS:**
STEPS:	**DECISION POINTS:**
FLOW CHART:	



DESIGN-BUILD-TEST

WHAT IS IT LIKE TO USE THE DESIGN-BUILD-TEST CYCLE LIKE A COMPUTER SCIENTIST?

FEATURING: **MIRAL KOTB**

Description: Students read about the design-build-test cycle, then they use it to create a paper prototype for their own mobile app idea.

LEARNING GOALS

Students will be able to...

1. Explain what happens in each stage of the design-build-test cycle.
2. Use the design-build-test cycle to create a paper prototype.

MATERIALS

1. Worksheet
2. Photo: Miral Kotb
3. Video: Miral Kotb - Made with Code
4. Additional blank paper and drawing materials for prototypes

STANDARDS

CSTA: 2A-AP-15
CCSS ELA: CCRA.SL.1
CCSS Math: MP3

VOCABULARY

Design-build-test cycle: A process for creating a product that involves designing what it should look like, building a draft, and then testing it to get feedback. This cycle can be repeated again and again to make a product better and better.
Prototype: An early version of a product that computer scientists use for testing and getting feedback, like a draft.

HOOK (5 MIN)

1. Ask: What is a problem or challenge that you face that could be fixed with a mobile app?
2. Set a timer for 1 minute while students write down as many ideas as they can think of.
3. Have students share out some of their ideas by filling in the sentence stem: "I'd build an app that helps with __________ by ___________."
4. Explain: Many great products are invented because they help someone solve a problem or they fill a need. Today, you'll get a chance to see what it's like to dream up a product and start innovating.
5. Distribute the worksheet and set a timer for 2 minutes while students read the sections "What is the Design-Build-Test Cycle?" and "What is a Prototype?"
6. Call on a few students to summarize what the design-build-test cycle is and why it's useful.
   - Suggested Answer: The design-build-test cycle helps computer scientists get feedback on their products and make them better.

WOMAN IN TECH SPOTLIGHT (5 MIN)

1. Play the Miral Kotb - Made with Code video.
   - Unplugged Alternative: Have students read the biography of Miral Kotb included on the worksheet.
2. Ask: When might Miral have used the design-build-test cycle when she was creating iLuminate?
   - Suggested Answers: Miral probably used the design-build-test cycle when she was creating the special costumes that her performers wear. She probably also used it when she was creating different kinds of visual effects for her shows, to test out which ones were the most exciting.

ACTIVITY (25 MIN)

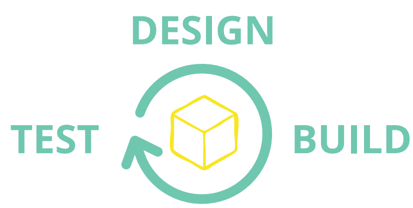
1. Give students a moment to choose one of the app ideas they came up with earlier to turn into a prototype.
2. Set a timer for 5 minutes while students fill out the Design section of the worksheet. During this time, students should make rough sketches for how they might design their apps. Encourage students to try out lots of different designs to see which they like best. They'll flesh out a single idea in more detail in the next section.
   - Students might feel nervous and claim they can't draw. Let them know that this activity isn't about creating a work of art; it's about getting their ideas across as easily as possible.
3. Explain: Now that you've had some time to try out some different designs, it's time to choose your best sketches and use them to build a paper prototype.
4. Set a timer for 15 minutes while students use the Build section of the worksheet to draw out a paper prototype for their app.
   - Have extra paper handy in case students need more than the three screens provided.
   - If you have more time, consider having students build a more interactive paper prototype, like the one shown in this video. This will let students get more accurate feedback in the Test phase.
5. Have students find a partner for the Test phase. Explain to them that providing feedback to teammates is an important skill for computer scientists, and something they're going to now practice!
6. Set a timer for 5 minutes. In this time, the first partner should demo their prototype and answer any questions the second partner has about how it works. Then the second partner should give feedback on the prototype, using the format of Glows and Grows. The first partner should take notes using the Test section of the worksheet.
   - Glows: Things that are working well.
   - Grows: Ways the product could be improved.
7. Set a timer for 5 minutes. Have the students switch roles so that the second partner has a chance to demo their prototype and get feedback.

DEBRIEF (5 MIN)

1. Set a timer for 1 minute. Have students write down what changes they might make to their prototypes if they were to do another round of design-build-test.
2. Ask for volunteers to share out their thoughts on the design-build-test cycle. How did it feel to use it? What did they like? What did they dislike?
3. Set a timer for 1 minute. Have students turn to a new partner and discuss the following: How might you apply the design-build-test cycle to other parts of your life? Why might it be useful to use in that situation?
4. Have a few students share out what they talked about with their partners. The best student answers will emphasize that the design-build-test cycle is helpful because it lets you work in smaller chunks and get feedback early, which lets you make adjustments quickly without wasting a lot of effort.
5. Tell students that they can use this strategy in any of their classes, and dive even deeper into it by taking a computer science class when it's available!

WHAT IS THE DESIGN-BUILD-TEST CYCLE?

Just like how a good author will go through multiple drafts when writing a book, a good computer scientist will build multiple versions of a product before it's finished. One process that computer scientists can use to build a product is the **design-build-test cycle**. In this cycle, computer scientists create a product by moving through three different stages:



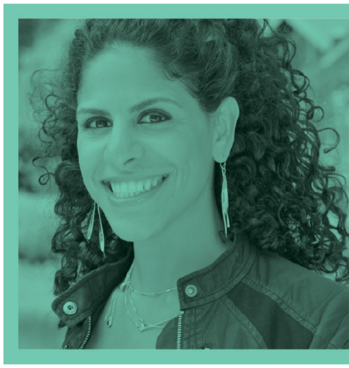
- **Design:** Deciding what the product should look like.
- **Build:** Creating a version of the product.
- **Test:** Getting feedback on the product from other people.

Once you finish the test phase, take what you've learned and use it in your next design phase. For example, in your tests you might learn that your users have a hard time finding a particular feature. So you might start a new design stage where you tweak your product's layout to improve the user's experience. This cycle can be repeated again and again to make a product better and better.

WHAT IS A PROTOTYPE?

A **prototype** is an early version of a product, like a first draft of an essay. Computer scientists will often start off by building a prototype, because it lets them quickly put something together so they can get feedback from actual users. It's better to use a prototype to test than a finished product, because if users don't like something, it's easier to make changes to a prototype. Think about a time when you wrote a paper for class. Wouldn't it be easier to make changes to your first outline than it would be to change your final draft?

MIRAL KOTB



Miral Kotb was born in Egypt, and she and her parents moved to the United States when she was 2. When Miral was 9, she built her first computer program. She went on to earn a degree in computer science at Columbia University while studying dance at Barnard College at the same time. After college, she started working as a software engineer, but she dreamed of combining her love of dance with her love of technology.

Then, in 2009, when Miral was working as an engineer on mobile apps, she had an idea. She imagined a performance where dancers wore costumes with lights that could be wirelessly turned on and off in time with the music. This idea led Miral to create iLuminate, a dance group that performs in special light-up suits to create visual illusions. As the company got started, Miral and her team built different prototypes of the costumes until they found one that worked just right. Miral likes that iLuminate brings together people from different backgrounds - programmers, dancers, costume designers - to create something magnificent!

When might Miral have used the design-build-test cycle when she was creating iLuminate?

NAME: ______________________

DATE: ______________________

INSTRUCTIONS: Choose a challenge or problem you face that could be solved with a mobile app. Use the space below to sketch some ideas for the design of your app. Don't worry about being perfect - just get your ideas down quickly! You'll have a chance to refine your ideas in the next section. As you sketch, consider the following questions:

- What is the main thing your app needs to do? What's the simplest way to help the user do that?
- What information goes on each screen of your app? How does the user move from one screen to the next?
- What buttons does your app need? What happens when the user clicks them?

CHALLENGE:

SOLUTION:

DESIGN:

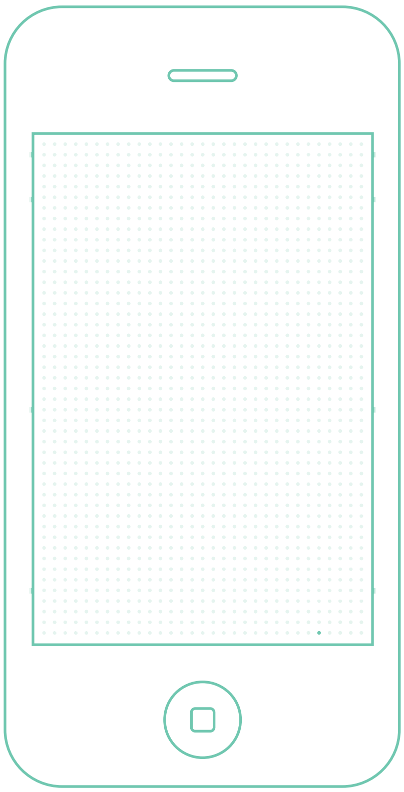
NAME: ______________________

DATE: ______________________

INSTRUCTIONS: Use the space below to build a paper prototype of your app.

BUILD:

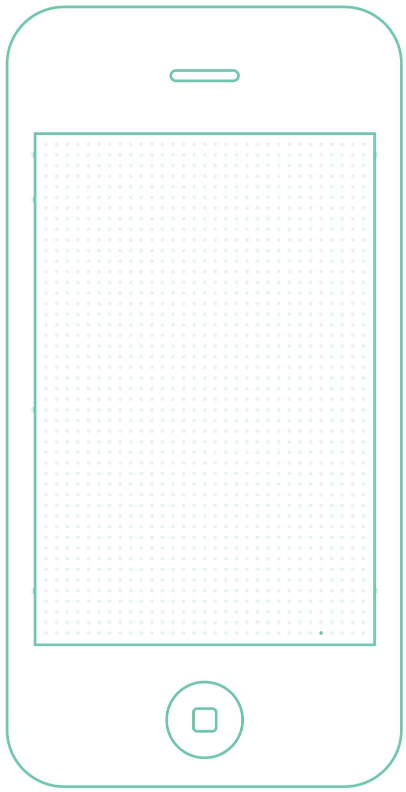
Screen: ______________________



Notes:

____

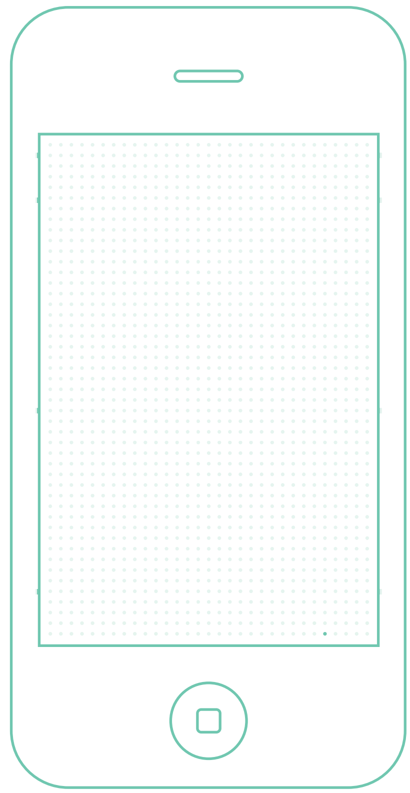
Screen: ______________________



Notes:

____

Screen: ______________________



Notes:

____

NAME: ______________________

DATE: ______________________

INSTRUCTIONS: Use the space below to keep track of the feedback you receive from your classmates.

GLOWS	**WHAT IS WORKING WELL?**
GROWS	**WHAT MIGHT YOU MAKE IT WORK EVEN BETTER?**

Based on the feedback you received, what changes might you make to your product if you were to do another round of design-build-test?

How might you apply the design-build-test cycle to other parts of your life?

DEBUGGING

HOW DO COMPUTER SCIENTISTS
FIND AND FIX PROBLEMS?

FEATURING: **GRACE HOPPER**

UNITED STATES NAVAL OFFICER, MATHEMATICIAN

Description: Students learn about identifying bugs, practice finding and fixing different types of bugs, and discuss the difference between types of bugs and strategies for debugging.

LEARNING GOALS

Students will be able to...

1. Define debugging.
2. Debug a set of instructions.
3. Name at least two types of bugs.

MATERIALS

1. Correct folding pattern video
2. Paper
3. Origami paper (optional)
4. Grace Hopper Article
5. COBOL Code compiler (optional)
6. Grace Hopper video

STANDARDS

CSTA: 1B-AP-15
CCSS ELA: CCRA.R.5
CCSS Math: MP3

VOCABULARY

Bug/error: a problem in computer code or a machine
Debug/debugging: the process of finding and fixing problems
Run-time error: a bug that happens after a program has started
Compile-time error: a bug that happens when reading instructions, before the program has started
Logic error: a bug where the steps do not lead to the correct outcome
Syntax error: a bug where the format of an instruction is wrong

HOOK (10 MIN)

1. Share today's learning goals with the students.
2. **Explain**: We're going to start today by creating origami boxes! Let's start by getting at least three or four square sheets of paper -- just in case you have to start over.
3. Hand out three to four square sheets (or three rectangles that they need to make into squares using the instructions) to each student.
4. **Set a timer for 5 minutes**. Ask students to use the Origami Instructions worksheet to make a box. Encourage them to try twice.
    - **Note**: The instructions are purposefully faulty. Students won't be able to make a box. If students find the error as they follow the instructions, tell them to keep going and see what happens.
5. **Explain**: It looks like you have a **bug** in your instructions! Try comparing your instructions with these to see if you can spot the bug.
6. Show students the correct instructions and ask them compare and find the incorrect instructions.
7. **Ask**: Were you able to find the mistake(s)? How did you find it?
8. **Explain**: In computer code, a mistake is called a **bug**. You will spend the rest of the activity building your **debugging** skills. Debugging is the art of finding and fixing problems, most often in computer code. Debugging skills apply to any type of project, whether it's finding the miscalculation in a math problem or making sure a logical argument is supported with relevant statements.

WOMAN IN TECH SPOTLIGHT (5 MIN)

1. Play the Grace Hopper Video.
    - **Unplugged Alternative**: Have students read an article about the Grace Hopper.
2. **Explain**: Grace Hopper created the first **compiler**. A compiler takes human-readable instructions and turns them into ones and zeros for a machine. She was a technical consultant on a committee which created a computer language called COBOL*, a language inspired by her compiler.
    - **CO**mmon **B**usiness-**O**riented **L**anguage
3. Write the following COBOL code on the board (or project it: COBOL):

```
IDENTIFICATION DIVISION.
        PROGRAM-ID. hello.
PROCEDURE DIVISION.
        DISPLAY “Hello World!”.
STOP RUN.
```

4. **Ask**: Can you guess what this COBOL code might do?
    - **Extension**: What would happen if you removed the “PROCEDURE DIVISION.”?

ACTIVITY (25 MIN)

Part 1: Short Challenges (10 min)

1. Pass out the Debugging Worksheet Part 1.
2. Split students into groups of 4 to 6.
3. **Explain**: Let’s do the first example together.
4. Have a student read the directions and the rules, then complete the first example as a class on the board.
5. **Explain**: In your groups, complete the mini-challenge. Remember to follow the rules from the example challenge! You can only remove or edit instructions -- no adding new instructions or making up your own.
6. **Set a timer for 5 minutes** as students complete the rest of part one.
7. **Ask**: Now that you have tried out a small challenge, let’s talk strategies. What kinds of bugs did you find in the instructions? Where there any similar bugs across mazes?
    - **Suggested Answers**:
        - Some instructions didn’t need to be there. Some bugs just had wrong numbers.
        - Some instructions weren’t about maze solving.
        - Some instructions were spelled incorrectly.
        - Some instructions didn’t do math correctly.

Part 2: The Main Challenge (10 min)

1. Pass out the Types of Bugs definitions.
2. **Explain**: Read about the types of bugs you can find and then in your groups, fill in the notes section with examples from the mini-challenge of a syntax error and a logic error.
3. **Explain**: Now that we've identified some types of bugs, can you do the next challenge?
4. Pass out the Debugging Worksheet Part 2.
5. **Set a timer for 8 minutes**. Students should set about debugging the final challenge. As you circulate the classroom, remind them to write the names of the type of bug they found and ask them to consider how they figured it out so they can use that same strategy in the future!

Close Out (5 min)

1. **Explain**: Let's solve the final challenge! Everyone reset your challenge. Every time I say go, you should do one instruction. Call out "Bug" if you edited or removed the step, and what type of bug it was. For example, if you removed an instruction that was not in English, you should call out "syntax error!"
2. Run through everyone's solutions by going one step at a time through everyone's instructions at the same time. Students should call out the names of the error as you run through instructions.

DEBRIEF (5 MIN)

1. **Explain**: You've all learned to put on a debugger's hat! How was it?
2. **Reflect**: Ask students to reflect on the questions at the end of the activity for a minute or two.
   - What strategies did your group use to debug? How did you collaborate or split up the work?
   - Think about bugs you found. Were some easier to find than others? What made them easier to find?
3. **Share**: Have a few students share out their responses with the class.
4. **Explain**: Debugging is something computer scientists do all the time. Outside of Computer Science, debugging is used by editors to copy-edit and fact-check novels and stories; by scientists to check the accuracy of scientific studies; and even by mathematicians to verify math proofs.

Grace Hopper is a mathematician and a rear admiral in the U.S. Navy. She was a pioneer in developing computer technology, helping to devise UNIVAC I, the first commercial electronic computer, and naval applications for COBOL (common-business-oriented language).

After graduating from Vassar College in 1928, Hopper attended Yale University. She taught mathematics at Vassar before joining the Naval Reserve in 1943. She became a lieutenant and was assigned to the Bureau of Ordnance's Computation Project at Harvard University (1944), where she worked on Mark I, the first large-scale automatic calculator and a precursor of electronic computers. She remained at Harvard as a civilian research fellow while maintaining her naval career as a reservist. After a moth infiltrated the circuits of Mark I, she coined the term bug to refer to unexplained computer failures.

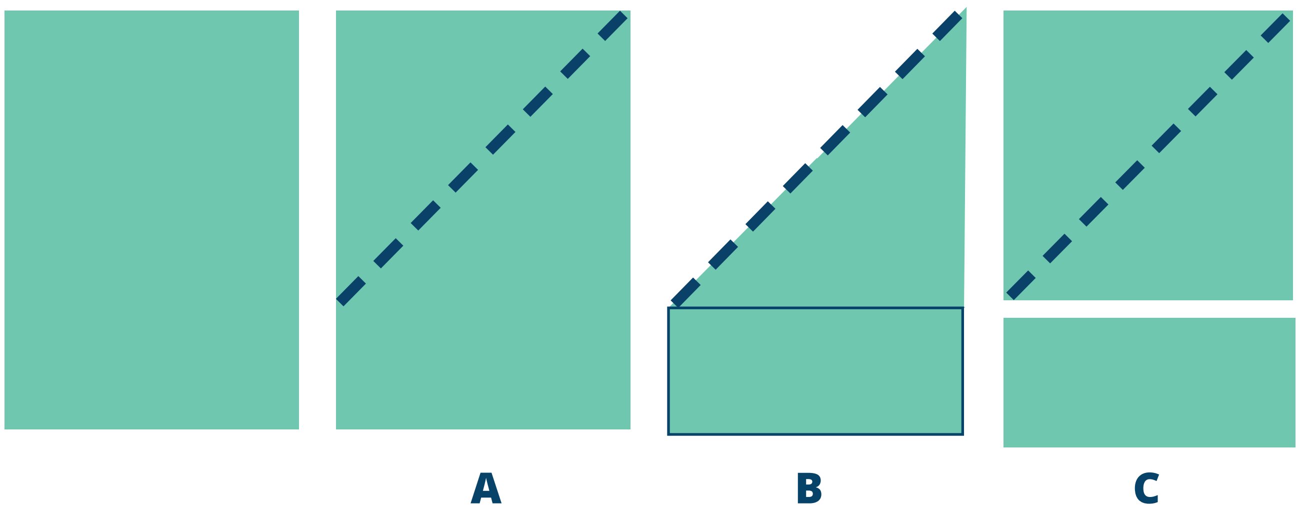
In 1949 Hopper joined the Eckert-Mauchly Computer Corp., where she designed an improved compiler, which translated a programmer's instructions into computer codes. She remained with the firm when it was taken over by Remington Rand (1951) and by Sperry Rand Corp. (1955). In 1957 her division developed Flow-Matic, the first English-language data-processing compiler. She retired from the navy with the rank of commander in 1966, but she was recalled to active duty the following year to help standardize the navy's computer languages. At the age of 79, she was the oldest officer on active U.S. naval duty when she retired again in 1986.

ORIGAMI INSTRUCTIONS

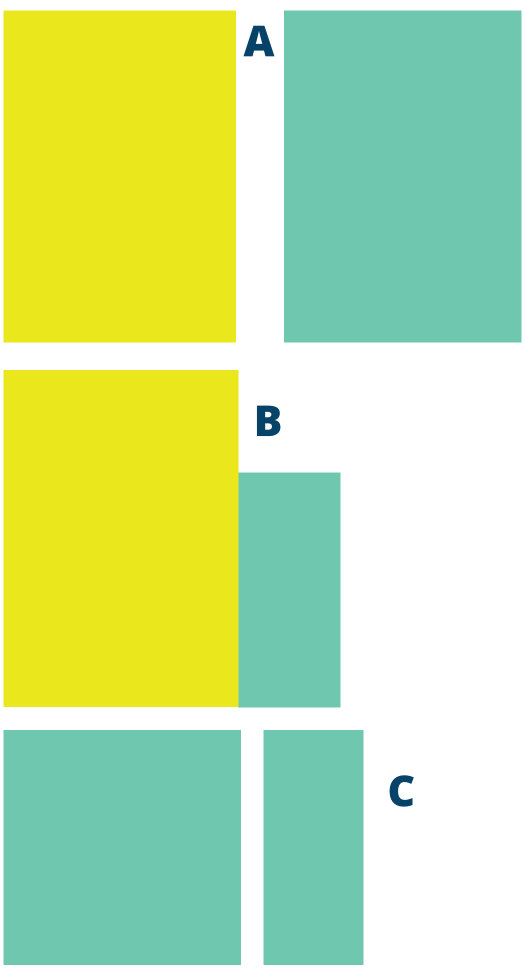
Rectangle to Squares Instructions:

Source: https://www.origami-resource-center.com/cut-a-square.html

Method 1:



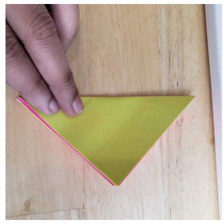
Method 2:



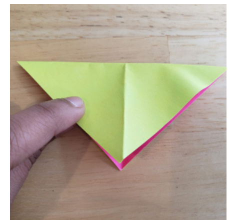
ORIGAMI BOX INSTRUCTIONS

*__Note__: For every fold, make sure you push hard to make a nice visible crease.

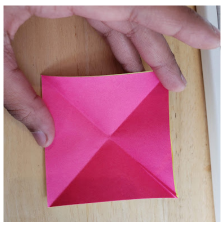
Face the color you want your box to be towards you



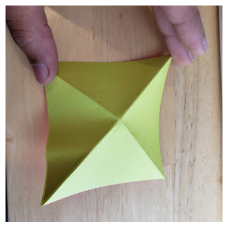
Fold in half diagonally*



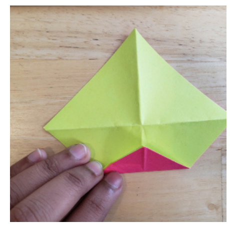
Open and fold again in half diagonally on the other diagonal



You should have two diagonals folding towards you

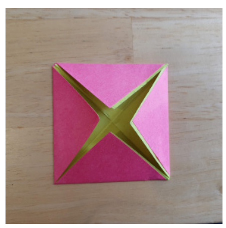


Turn it over



Fold a corner toward the center

Keep going, fold all the corners toward the center



You should have all four corners folded inwards

Fold the bottom to the middle

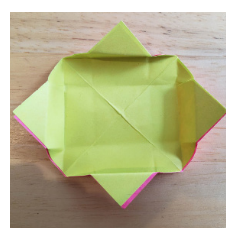
Fold the top to the middle so that you have two halves folded to the center

Open

Turn 90 degrees

Repeat the same folds: fold the bottom to the middle

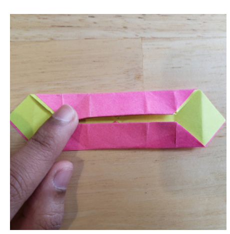
Fold the top to the middle so that you have halves folded to the center



Open

Fold the bottom to the middle again

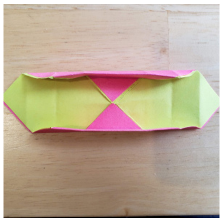
Finish the fold and crease well



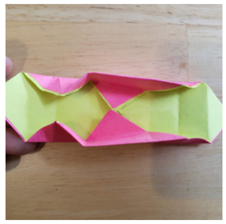
Fold the top to the middle

ORIGAMI BOX INSTRUCTIONS

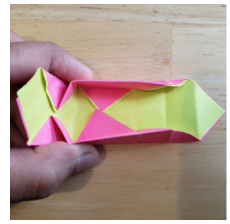
*__Note__: For every fold, make sure you push hard to make a nice visible crease.



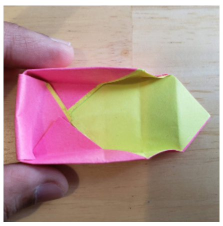
Open part ways. These will be the sides of your box



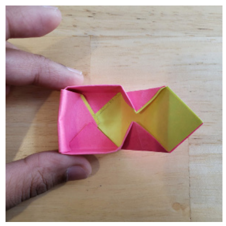
Push the left side towards the middle. It will want to lift up and fold as shown



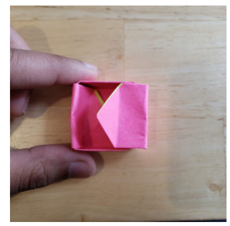
Crease as needed and finish the lift



Fold the downwards and push the end into the box



Repeat on the right side by pushing it towards the middle as it lifts up



Push it downwards into the box

The finished product!

NAME: ______________________________

DATE: ______________________________

SETUP: You were working on an assignment late last night. As you review your homework in the morning, you realize you wrote some errors into the instructions because you were sleepy. Can you **debug** all the **bugs**?

The goal of the game is start at the **start** point ☞, pick up all the stars ★, and get to the **end** point ⚐ by finding and fixing all the **bugs** in the instructions. The direction the finger points is the direction you're facing when the game starts.

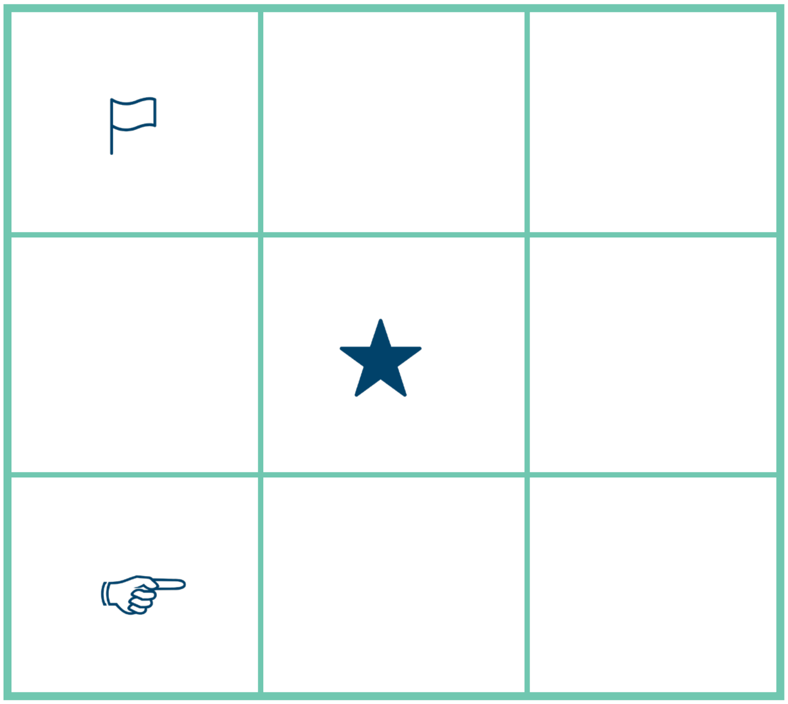
RULES:

1. You can only ***edit*** or ***remove*** instructions. Editing a step means replacing any [values] in the instruction with a different value or fixing a spelling error.
2. You may ***not*** add instructions of your own.
3. Your character ***must*** start at the start point facing the direction of the pointer: ☟ ☞ ☜
4. If your character gets to the end point, the game ends ***even if*** there are more instructions.
5. ***Only*** these instructions are allowed:
   - move [1 to 5] steps forward
   - repeat [1 to 5] times: ... stop repeat
   - turn [right/left/around]
   - pick up star

NAME: ____________________

DATE: ____________________

EXAMPLE CHALLENGE



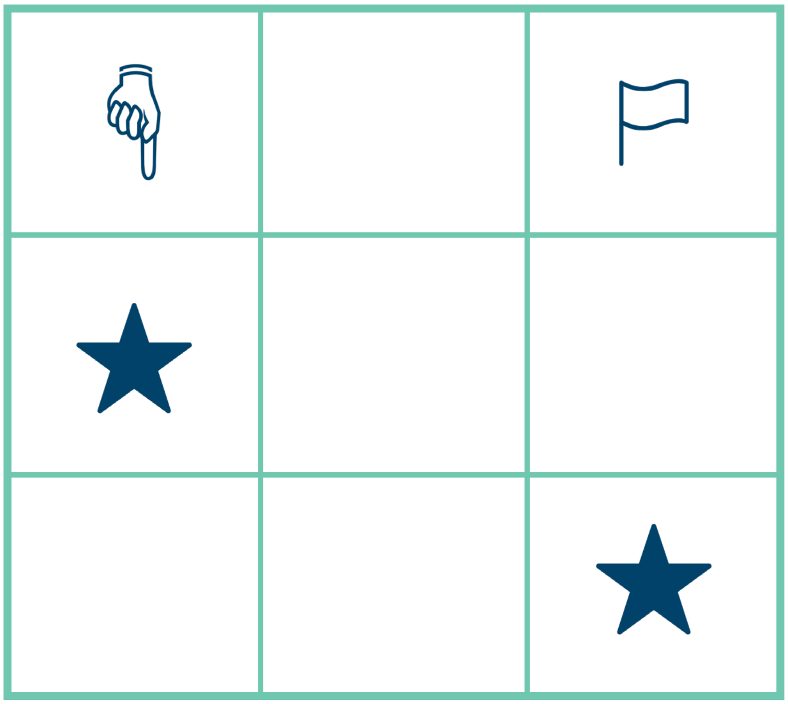
Incorrect Instructions	*Debugged Instructions*
Repeat **1** times:	
move **1** step forward	
turn **left**	
move **1** step forward	
pick up star	
turn **around**	
stop repeat	

Hint: There are two bugs here.

NAME: ______________________

DATE: ______________________

MINI CHALLENGE



Incorrect Instructions	*Debugged Instructions*
move 1 steps forward	
pick up the star	
moooove 1 steps forward	
turn **here**	
repeat 2 times:	
move 2 steps forward	
ppkccc up star	
turn **left**	
reach for the skies	
stop repeat	
turn **left**	

Hint: There are five bugs here.

TYPES OF BUGS

There are two major types of bugs: **compile-time** and **runtime** errors. Compile-time errors are bugs that happen when a computer compiles* instructions into a program that you can click on and run. Runtime errors are bugs that happen once you've clicked on a program, and it starts running.

* Compiling is the process of turning a list of instructions into a program. It's how a game goes from being lines of code to an icon on your screen!

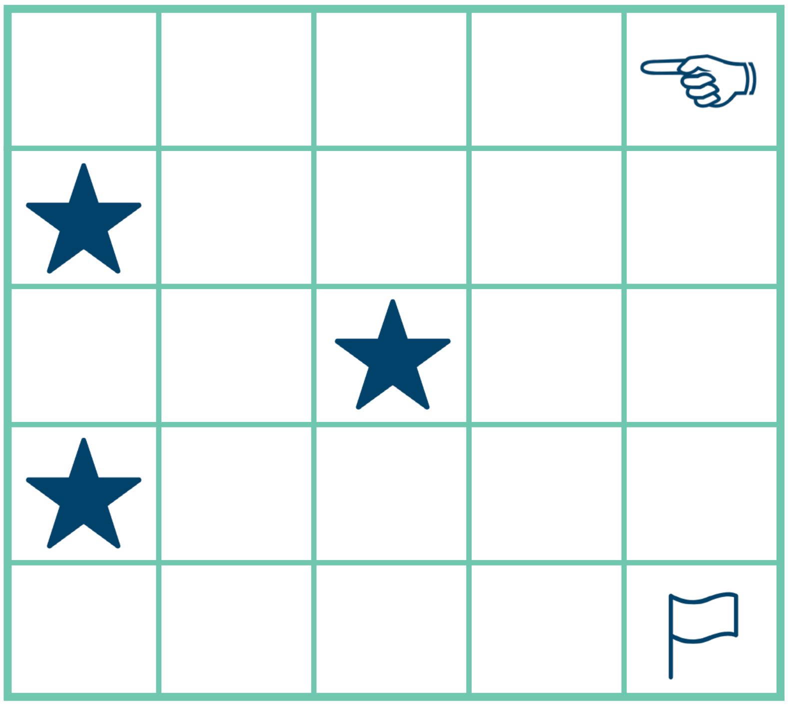
EXAMPLES

Bug	*Example*	*Notes*
Syntax Error (compile-time error) A bug where the format of an instruction is wrong.	• Spelling something incorrectly • Using another language • Using an instruction that doesn't exist • Using the wrong punctuation or missing punctuation	
Logic Error (runtime error) A bug where the steps do not lead to the correct outcome.	• Doing an action too many times or too few times • Using an incorrect instruction • Using too many instructions or too few instructions • Using the incorrect order of instructions	

DEBUGGING WORKSHEET - PART 2

NAME: ______________________________

DATE: ______________________________



Incorrect Instructions	*Debugged Instructions*	*Type of bug (syntax or logic)*
move **7** steps		
turn **left**		
move **1** step		
pick up staaarz		
move **2** steps		
pick up star		
turn right		
twirl in place		
move **1** step		
move **1** step		
turn **right**		

DEBUGGING WORKSHEET - PART 2

NAME: ______________________

DATE: ______________________

Incorrect Instructions	*Debugged Instructions*	*Type of bug (syntax or logic)*
repeat **2** times		
move **1** steps		
pick up star		
draw a line		
stop!! that!! repeat!		
turn **'ight**		
move **2** steps		
gooooooal		

REFLECTION

NAME: ______________________

DATE: ______________________

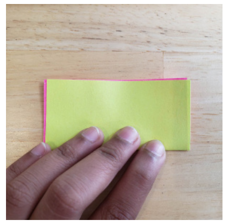
What strategies did your group use to debug? How did you collaborate or split up the work?

Think about bugs you found. Were some easier to find than others? What made them easier to find?

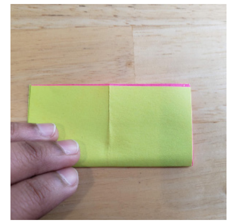
ANSWER KEY (TEACHER ONLY)

Correct Origami Instructions

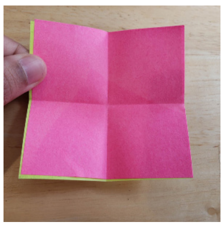
Face the color you want your box to be towards you



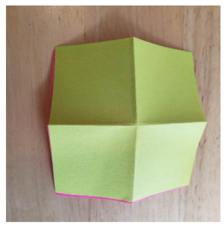
Fold in half lengthwise



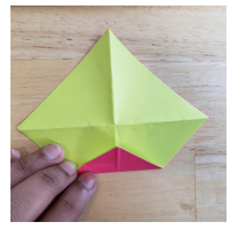
Open and fold again in half lengthwise on the other side



You should have two lengthwise folds towards you

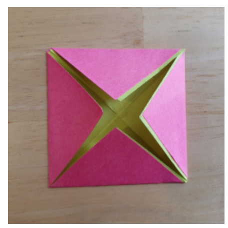


Turn it over



Fold a corner toward the center

Keep going, fold all the corners toward the center



You should have all four corners folded inwards

COBOL Code Answers:

```
IDENTIFICATION DIVISION.
        PROGRAM-ID. hello.
PROCEDURE DIVISION.
        DISPLAY "Hello World!".
STOP RUN.
```

Can you guess what this COBOL code might do?

Answer: It displays "Hello World" on your screen!

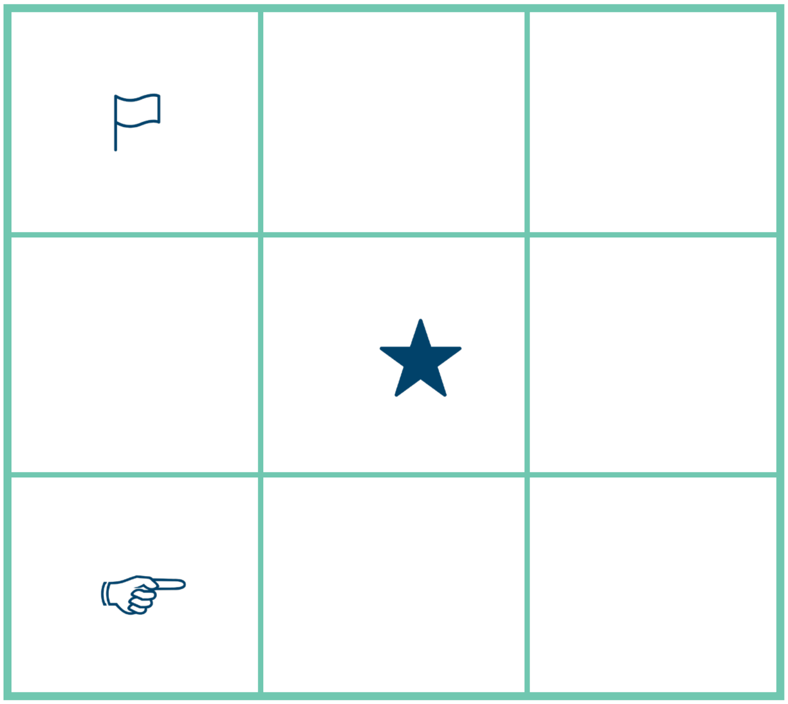
What would happen if you removed the "PROCEDURE DIVISION."?

Answers: It won't run! / There will be an error.

NAME:

DATE:

EXAMPLE CHALLENGE (suggested answer):



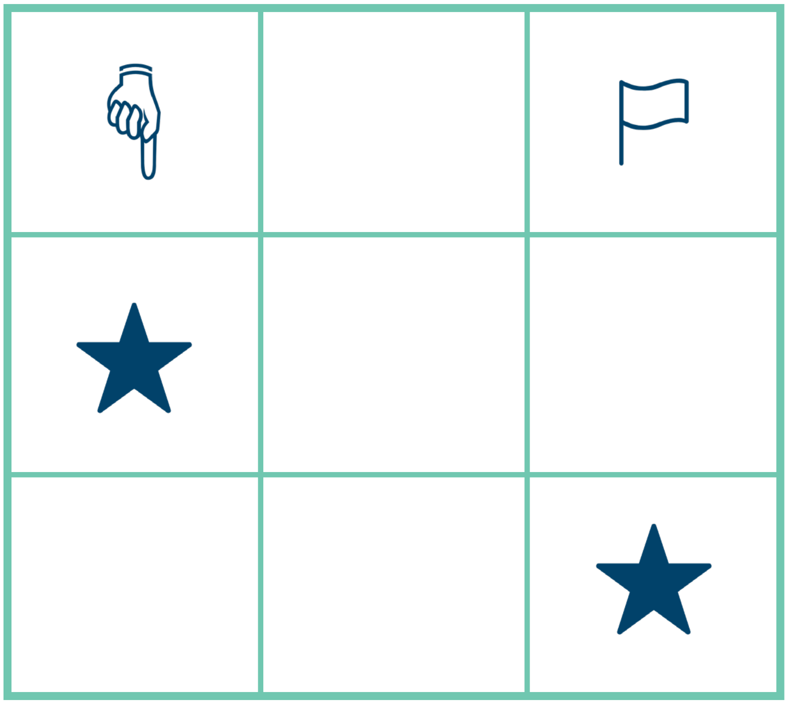
Incorrect Instructions	*Debugged Instructions*
Repeat 1 times:	Repeat **2** times:
move 1 step forward	move 1 step forward
turn **left**	turn **left**
move 1 step forward	move 1 step forward
pick up star	pick up star
turn **around**	**-- delete instruction --**
stop repeat	stop repeat

Hint: There are two bugs here.

NAME: ______________________

DATE: ______________________

MINI CHALLENGE (suggested answer)

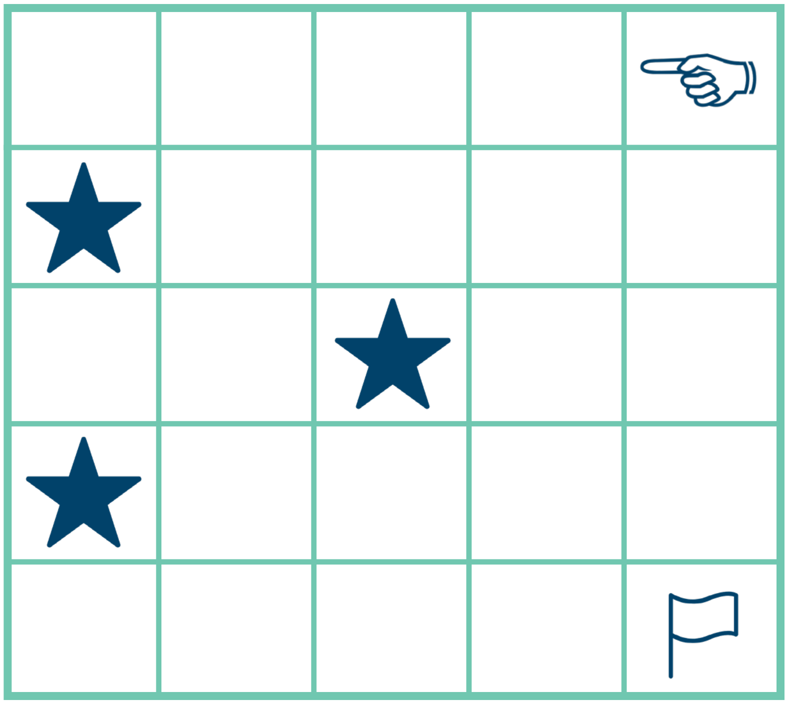


Incorrect Instructions	*Debugged Instructions*
move 1 steps forward	move 1 steps forward
pick up the star	pick up the star
moooove 1 steps forward	**move** 1 steps forward
turn **here**	turn **left**
repeat 2 times:	repeat 2 times:
move 2 steps forward	move 2 steps forward
ppkccc up star	**pick** up star
turn **left**	turn **left**
reach for the skies	**-- delete instruction --**
stop repeat	stop repeat
turn **left**	**-- delete instruction --**

Hint: There are five bugs here.

DEBUGGING WORKSHEET - PART 2

Note: Students may come up with other solutions.



Incorrect Instructions	*Debugged Instructions*	*Type of bug (syntax or logic)*
move **7** steps	move **4** steps	Logic
turn **left**	turn **left**	
move **1** step	move **1** step	
pick up staaarz	pick up **star**	Syntax
move **2** steps	move **2** steps	
pick up star	pick up star	
turn **right**	turn **around**	Logic
twirl in place	**-- delete instruction --**	Logic
move **1** step	move **1** step	
move **1** step	**-- delete instruction --**	Logic
turn **right**	turn **right**	

DEBUGGING WORKSHEET - PART 2

NAME: ______________________

DATE: ______________________

Incorrect Instructions	*Debugged Instructions*	*Type of bug (syntax or logic)*
repeat **2** times	repeat **2** times	
move **1** steps	move **2** steps	Logic
pick up star	pick up star	
draw a line	**-- delete instruction --**	Logic
stop!! that!! repeat!	**stop repeat**	Syntax
turn **'ight**	turn **right**	Syntax
move **2** steps	move **2** steps	
gooooooal	**-- delete instruction --**	Syntax

HARDWARE + SOFTWARE

HOW DO HARDWARE AND SOFTWARE WORK TOGETHER TO IMPACT THE WORLD?

FEATURING: **KATHERINE JOHNSON**

Description: Students learn the difference between hardware and software in abstract and practical terms and understand how they interact with one another through an interactive game.

LEARNING GOALS

Students will be able to...

1. Explain the difference between hardware and software
2. Model how hardware and software interact with each other

MATERIALS

1. Worksheet
2. Video: Hardware and Software Explained
3. Video: Katherine Johnson - Biography
4. Additional blank paper and drawing materials for prototypes

STANDARDS

CSTA: 1B-CS-02
CCSS ELA: CCRA.SL.4
CCSS Math: MP6

VOCABULARY

Hardware: a physical object. In the case of a computer, hardware is the monitor, mouse, keyboard, and all of the other parts that you can actually see or touch.
Software: the instructions that tell the hardware what to do. For a computer, the software includes programs and algorithms that allow a computer to run and allow a user to access information.

HOOK (10 MIN)

1. Share today's learning goals with the students and read out the vocabulary words and their definitions for the class.
    - **Tip**: Write these words and their definitions up before the lesson.
2. **Explain**: Giving and receiving instructions is a common part of everyday life. As a teacher, I give you instructions often. Sometimes I give instructions with my voice, sometimes with written directions, and sometimes with hand signals. There are also tools that we use to give instructions to other people and objects. For example, a recipe may give instructions on how to create a meal.
3. **Set a timer for 2 minutes**: Have students brainstorm other tools that control or give instructions to other objects.
    - **Examples**: Garage opening remote, buttons disarming or arming alarm system in homes, video games and controllers, etc.
4. **Analyze and Share Out**: Which items on your list are hardware, and which ones are software?
    - **Hint**: The one that gives the instructions or controls the other one is the software. The one that it is controlling is the hardware.
5. **Watch this video:** Hardware and Software Explained
    - **Unplugged alternative**: Distribute the worksheet and have students read the sections "What is Hardware?" "What is Software?" and "How do they interact?"
6. Call on a few students to summarize how hardware and software work together.
    - **Suggested Answer**: Hardware is a physical object, and software is the set of instructions that allows it to work and complete tasks.

WOMAN IN TECH SPOTLIGHT (5 MIN)

1. Play the Katherine Johnson - Biography video.
    - **Unplugged Alternative**: Have students read the biography of Katherine Johnson included on the worksheet.
2. **Explain**: This video shared a history of Katherine Johnson's accomplishments as a mathematician and computer scientist.
3. **Ask**: How did Katherine connect hardware and software when she was doing the calculations to provide accurate instructions for the moon-landing?
    - **Suggested Answers**: Katherine connected the two by providing instructions to ensure that the moon-landing was done successfully. The instructions were the software to make sure that the hardware (actual spacecraft) landed on the moon as it was supposed to.

ACTIVITY (25 MIN)

Part 1: Design Your Spacecraft (10 minutes)

1. Split students into groups of 2, and pass out the Part 1 Worksheet to each group.
2. Ask one student to read out the instructions at the top of the worksheet.
3. **Set a timer for 2 minutes**: As a class, brainstorm parts of a spacecraft that are important to include in the drawings. Focus on what needs to exist so that the astronaut can receive information and act on those instructions to fly the spacecraft.
    - Suggested Answers: Windows, fuel tank, seats, doors, wings, landing gear
4. **Set a timer for 5 minutes**: Ask each group to draw their spacecraft in the space on the worksheet.
    - **Note**: Students might feel nervous and claim they can't draw. Let them know that this activity isn't about creating a work of art; it's about getting their ideas across as best as they can.
5. **Prepare**: While they are working on their drawing, create a maze for the students to follow to complete their space mission.
    - **Tip**: You can do this by rearranging the desks, creating a path through the classroom with tape, or drawing or projecting a picture of a maze on the board. Include obstacles such as desks or other objects as asteroids that they have to avoid during their mission.

Part 2: Program Your Spacecraft (15 minutes)

1. Pass out the Part 2 Worksheet to each group.
2. **Explain**: Your mission will be to move your spacecraft through the maze and land it successfully.

Part 2: Program Your Spacecraft (continued)

3. As a class: Have one student read out the instructions for part two.
4. **Explain**: For hardware and software to interact, directions need to be exact. The control center must tell the spacecraft exactly what to do, and the spacecraft must only do what they are told by the control center.
5. **Set a timer for 2 minutes:** Have students plan out their route and brainstorm instructions to complete the mission.
6. **Go!** Have three of the groups go through the maze, one by one, while allowing the rest of the class to watch.
    - Remind students that only the control centers should be speaking, and "programming" the spacecraft by providing instructions on what their teammate should do. If students notice someone moving without instructions, ask them to call it out to engage class participation and make sure that the hardware and software are working together properly.
7. Now that students have gone through once, tell them that they are going to do this another time but without any verbal instructions.
8. **Set a timer for 5 minutes**: Allow pairs of students to work together to create a system for nonverbal communication to get through the maze. This could be based on hand or bodily gestures, eye movements, and more.
    - **Explain**: Most hardware and software interact nonverbally, so this will be more realistic for providing instructions in a coded way!
9. **Go!** Have three of the groups go through the maze, one by one, while allowing the rest of the class to watch.
    - Remind students that only the control centers should be speaking, and "programming" the spacecraft by providing instructions on what their teammate should do. If students notice someone moving without instructions, ask them to call it out to engage class participation and make sure that the hardware and software are working together properly.

DEBRIEF (5 MIN)

1. **Explain**: You successfully programmed hardware and saw how software and hardware interact, just like computer scientists do every day!
2. **Ask**: When have you seen hardware and software interact well? How about a time where it did not?
    - **Note**: Prompt students with examples of hardware and software you've explored throughout this lesson, including something that gives instructions to or controls something else.
3. **Close-out:** Refocus on the essential question for this lesson: *How do hardware and software work together to impact the world?* Allow 2-3 students to share out reflections about this question.
    - **Example answer:** Hardware and software working together is why we are able to have access to the internet and knowledge. It is also why we are able to do things like post on the internet, in which case we are acting as the software controlling hardware.

WHAT ARE HARDWARE + SOFTWARE?

1

WHAT IS HARDWARE?

Hardware refers to the physical elements of a computer. Examples of hardware in a computer are the keyboard, the monitor, the mouse and the central processing unit (CPU). However, most of a computer's hardware cannot be seen because it is inside the computer. Hardware performs tasks after receiving instructions from software.

WHAT IS SOFTWARE?

Software is made up of instructions, or electrical signals that tell hardware what should be done and when. In general, these instructions are referred to as programs, because they "program" the hardware to understand what it needs to do.

HOW DO HARDWARE + SOFTWARE INTERACT?

At the most simple level, the CPU (Central Processing Unit) of the computer controls all of the other parts of the computer. It has different types of circuits inside of it that instruct the hardware to perform certain tasks based on commands or code that it receives from software. The software tells the CPU what to do, and then the CPU makes it happen in the hardware.

Hardware and software always work together. Without software, the physical parts of a computer would have no purpose or function. However, without hardware to complete the tasks, software would also be useless.

KATHERINE JOHNSON

Katherine Johnson's love for math is evident in her academic and professional pursuits and is what makes her a stellar computer scientist. She is the third African American to get a PhD in Mathematics, after which she taught within a black public school in Virginia in 1937. While she was teaching, she began to see how math and problem-solving were connected to several different fields. In 1962, she was called upon by NASA to execute computer programs by hand to make sure that the math was correct. Once she approved their calculations, they proceeded on their mission.

At 100 years old, Katherine said that her greatest contribution to the field was her calculations that made sure the commanding and service modules for Project Apollo's moon-landing were aligned. Her instructions and calculations allowed humans to reach the moon successfully for the first time ever! In 2015, Obama presented her with the Presidential Medal of Freedom, which is America's highest civilian honor. Her story was turned into a semi-biographical film called ***Hidden Figures.***

QUESTION: How did Katherine Johnson contribute to the moon-landing?

NAME: ______________________________

DATE: ______________________________

HARDWARE + SOFTWARE (Part 1)

INSTRUCTIONS: In pairs, design your own spacecraft in the area below. Don't worry about being perfect - the more important part of the exercise will be programming it so that it can successfully complete its mission.

As you sketch, consider the following questions:

- What does the spacecraft need to do? Which components of the ship do those tasks?
- What tools does the astronaut need to receive information about how to fly the spacecraft?

Pick one person on your team to be the astronaut flying the spacecraft, and the other person to be the control center that provides instructions for how and where to fly.

DESIGN:

NAME: ______________________

DATE: ______________________

HARDWARE + SOFTWARE (Part 2)

INSTRUCTIONS: Prepare your route! If you are the control center, think about what instructions you will give your astronaut to get through the maze and complete your mission. Use the space below to map out your path and instructions. Remember, avoid the obstacles to make sure that the astronaut and spacecraft remain safe throughout the mission!

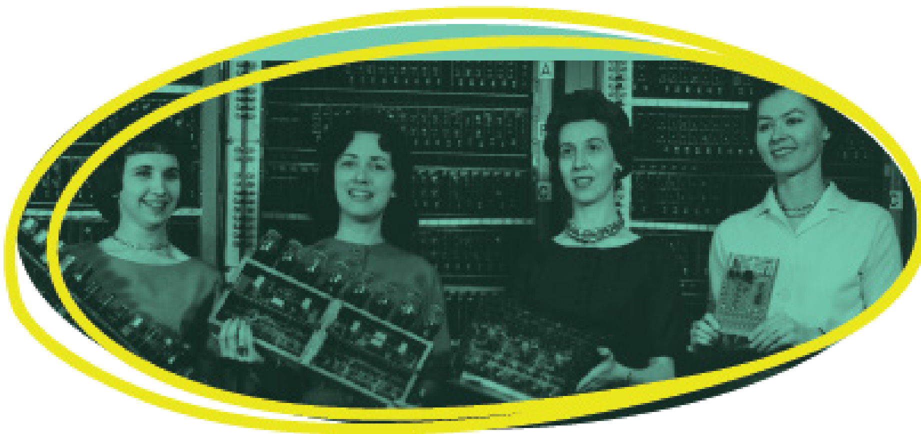
MAP IT

START→ **→FINISH**

QUESTION: Who was the hardware and who was the software? How did they work together to complete this mission?

PATTERN RECOGNITION

HOW DO COMPUTER SCIENTISTS USE PATTERN RECOGNITION TO SOLVE PROBLEMS?



FEATURING: **ENIAC Programmers**

Description: Students learn about pattern recognition, practice and create ciphers, and discuss how a computer can use pattern recognition.

LEARNING GOALS

Students will be able to...

1. Define Pattern Recognition.
2. Use pattern recognition on a cipher.
3. Name one way a computer can use pattern recognition.

MATERIALS

1. Pattern Recognition Brief
2. Pig Latin Worksheet
3. Answer Key
4. Code Breaking Worksheet
5. Scissors
6. Fastener
7. Cardboards
8. Video: ENIAC Programmers Project
9. Article: ENIAC Programmers

STANDARDS

CSTA: 2-NI-06
CCSS ELA: CCRA.R.1
CCSS Math: MP1

VOCABULARY

Pattern recognition: finding a pattern in a set of data
Cipher: a code; a way to change information so someone else can't understand it
Code/encode: convert information into a cipher
Decode: convert a cipher back into information

HOOK (10 minutes)

1. Share today's learning goals with the students and ask them to read sample text that has been **encoded** using Pig Latin.
2. Ask students to count off by threes.
3. **Set a timer for 2 minutes**. Ask students try to **decode** their assigned sentence on the Pig Latin Worksheet back into English.
    - **Extension**: If students finish early, encourage them to decode the remaining sentences back into English.
4. Ask students to share the answers they came up with.
5. **Set a timer for 3 minutes**. Ask students to turn to their elbow-partner and come up with 2 to 3 rules for Pig Latin together.
6. As a class, have students share out rules. How does Pig Latin work? What patterns emerged as they translated each sentence?
    - Focus on **Why/How** they came up with the rules and what patterns they identified in the sample. Use the Answer Key ONLY as a guide because answers will vary.
7. Highlight that students used **pattern recognition**. They used a sample sentence to figure out the rules and decode new messages using them.

HOOK (Continued)

8. Set a timer for 3 minutes. Have students read out loud the brief description of pattern recognition and its uses using the popcorn method. One person starts and picks the next person to read after each sentence.

9. Tell students that they will spend the rest of the activity building their pattern recognition muscle by using historic methods of encoding and decoding messages, as they already began doing with Pig Latin.

WOMAN IN TECH SPOTLIGHT (5 minutes)

1. Play the ENIAC Programmers Project trailer [1:10 - 3:35] video.
   - Unplugged Alternative: Have students read an article about the ENIAC Programmers.
2. **Explain**: This video talked about the women who were the first programmers of the ENIAC computer. Everything they did was new -- can you imagine being handed all the parts for a computer and given the instruction, "Now put it together and make it work?"
3. **Ask**: What are some differences you noticed between the ENIAC and computers today?
   - **Suggested Answers**: There are no screens on the ENIAC. There are a lot of cables everywhere. The women appear to be plugging in wires from different places. It's huge!

ACTIVITY (25 minutes)

Part 1: The Caesar Cipher (10 minutes)

1. Pass out the Code Breaking Worksheet. Have students cut out their **Caesar Cipher** encoder/decoders.
2. Ask a student to read out the rules for the Caesar Cipher.
3. Write "ENIAC" on the board.
4. Ask a student come up to the board while the class follows along and helps them encode the word: ENIAC using **1** rotation.
5. Split students into groups of 2 - 3 people.
6. **Set a timer for 3 minutes**. Each group should create their own Caesar Cipher using a number between 1 and 26. Tell them to translate the word ENIAC and two other 5-letter words of their choosing.
   - Example: ENIAC, HELLO, WORLD with 2 rotations is "GPKCE JGNNQ YQTNF"
7. **Set a timer for 5 minutes**. Students should pass only the three **encoded** words in a random order to the another group. Groups should try to figure out what the encoded words are using the knowledge that at least one translates to ENIAC.

Part 2: The Caesar Cipher with too much data (10 minutes)

1. **Explain**: In your groups, instead of creating a cipher with just three words, you will **encode** three to five long sentences with the Caesar Cipher.
   - **Note**: The sentences don't have to make sense together or make a paragraph. Any 3 - 5 sentences will do.
2. **Set a timer for 5 minutes**. Groups should create their ciphered sentences.
3. **Explain**: Before you give your cipher to another group, brainstorm some strategies for how you'll solve this next one. Strategies will help you break the ciphers faster!
4. **Set a timer for 2 minutes**. Students should strategize in their groups.
   - **Guiding Questions**:
     - Are there words that are easier to break than others? How might you decipher a one-letter or two-letter word?

6. Set a timer for 3 minutes. Groups should trade ciphers and try to break the new cipher.

- **Note**: Students are **not** expected to be able to break this cipher.

Part 3: Close out (5 minutes)

1. Close out the activity with the following Think-Pair-Share:
2. **Think**: Ask them to silently reflect on the questions at the end of the activity for a minute.
   - What strategies did you use to try to **decode** the cipher? How did you use these to decide if something was right or wrong?
     - Suggested Answers:
       - We converted one word at a time to see if it was a real word or a word in English.
       - We converted the smaller words first because those were the fastest to look at.
   - Could you have broken the second cipher with enough time? What if it was the length of a book? Why or why not?
3. **Pair**: Ask them to turn to an elbow partner and share their thoughts on the questions.
4. **Share**: Have a few students share out their thoughts on the questions with the class.

DEBRIEF (5 MIN)

1. **Explain**: Imagine you now had a computer to break the cipher. This computer processes information so quickly that the length of the message doesn't matter.
2. **Ask**: Let's brainstorm; what are some useful things a computer could do to help decode the cipher?
    - Guiding Question:
        - If the computer decodes one word using a specific rotation, how do you know if it's right?
    - Suggested Answers:
        - The computer should be able to tell if a word is in English or the word is in the English dictionary.
        - The computer should be able to tell if all the words are in English.
        - The computer should be able to tell how long a word is so it can look at the shortest words first.
        - The computer should be able to change the text by whatever rotation we're using.
        - The computer should be able to go through all the rotations in the Cipher Disc that you created.
3. **Close Out**: Congratulations! You all learned how to recognize a pattern and created strategies for how a computer could break codes. In Computer Science, a list of strategies that solve a problem is called an algorithm. Algorithms are one way computer scientists use pattern recognition to solve problems.
    - **Extension**: Close out by talking about how pattern recognition applies to the subject matter of your class by using the standards guide.

WHAT IS PATTERN RECOGNITION?

Pattern recognition is a way to find trends and make predictions. We use patterns in data to determine key features and predict outcomes.

HOW IS IT USED?

A computer software may track the things you buy. It can use this data to recognize patterns and predict your preferences.

Mathematicians look for patterns in groups of shapes to predict what they have in common. They predicted and proved that inside angles in a triangle add to 180 degrees.

Pattern recognition allows readers to understand how writers use grammar. For example, an active voice puts focus on a character. Passive voice puts the focus on an action. Knowing that readers recognize this can be used to change the mood in a passage.

Scientists use pattern recognition to analyze nature. For example, they study patterns in animals to group them by Kingdom, Phylum, and so on.

Historians classify governments by studying patterns in societies. Examples include democracies, monarchies, and so on.

ENIAC PROGRAMMERS

During World War II, advances in military technology had created a need for individuals called "computers" who could solve the lengthy and complicated equations needed to aim and target large ballistic weapons. As with most wartime positions, there were few qualified men who were available for the job. So in the 1940s, the U.S. Army recruited over 100 women to work as computers at the Army's Ballistic Research Labs in Philadelphia, Pennsylvania.

Yet even with dozens of America's most intelligent women working on the problems, doing the calculations by hand took far too long. In 1943, engineers John Mauchly and John Presper Eckert were given permission to pursue an experimental project — an electrical machine that could handle the computations in a matter of seconds. This machine, the Electrical Numerical Integrator and Computer (ENIAC), was completed in 1945 and is considered the world's first electronic computer. But unlike modern computers, it had no memory or ability to store the equations and processes needed to actually complete the calculations. A team of human computers who could complete the equations by hand were needed to hard-program these sequences into the ENIAC.

Six young women were chosen for the task: Frances "Betty" Holberton, Kathleen "Kay" McNulty, Marlyn Wescoff, Ruth Lichterman, Frances Spence, and Jean Jennings. The women had no precedent or manuals to follow and had to teach themselves the functions of the 30-ton machine, which was comprised of approximately 70,000 resistors, 10,000 capacitors, and 6,000 manual switches. In order for the ENIAC to work through the differential equations, the programmers had to break the bulky sequences into smaller, simplified steps, and literally hand-program the 1,800 square foot machine.

The result was groundbreaking. The six ENIAC women had successfully managed to cut the time-table on the ballistic equations from over 30 hours to mere seconds. In February of 1946, just months after the end of the war, the world's first electronic computer was revealed to the public.

But none of its programmers were present or named. Instead, male engineers Presper Eckert and Mauchly were elevated to fame and credited with the full creation and function of the ENIAC. Despite going on to make names for themselves in the industry, the female programmers and their contributions were over-looked for almost 70 years — at the 50th-anniversary celebration of the ENIAC project, the majority of the team wasn’t even invited.

In the 1980s, a young computer science major named Kathy Kleiman grew frustrated by the small number of women in her classes. While looking into the role of women in computer science, she came across the photos of the ENIAC project, which featured uncaptioned photos of the ENIAC programmers. According to Kleiman, when she asked a professor about the women, she was told they were likely “refrigerator ladies,” models who were posed with the machine to make it more alluring. This search for the pioneers of programming spawned a 20-year project called the ENIAC Programmers Project, as a way to promote the six women and their contributions.

What Kleiman and her collaborators found was heartening. While history was quickly working to forget them, the ENIAC women had continued to press forward with technological advances. Each woman went on to make her own mark in the field: Lichterman stayed with the ENIAC program for two years to train new programmers, while Holbertson and Jennings helped convert the ENIAC into a “stored-program” machine. Holbertson created the first program that allowed for sorting and storing large data files, and members of the team would later work to develop guidelines and standards for a universal programming language. Jennings spent the remainder of her career working to make computers more accessible and easier to use.

The ENIAC Programmers project paid off — not just as a way to memorialize the original programmers, but to promote the important work that women have done, and continue to do, for the field. This year marks 70 years since the first electronic computer was unveiled to the world, and although the programmers’ names were temporarily lost to history, they have now taken their rightful places in the hall of fame, inspiring the next generations of pioneers.

NAME: ______________________

DATE: ______________________

PIG LATIN WORKSHEET

INSTRUCTIONS: Decode the endcoded phrases below.

SAMPLE:
Encoded: Idday youway avehay away eatgray ayday?
Decoded: Did you have a great day?

Sentence 1: Erethay ereway onay reensscay onway ethay ENIACway, osay ethay omenway usedway aperpay otay eepkay acktray ofway everythingway.

Sentence 2: Ethay irstfway "omputerscay" ereway omenway howay idday athmay alculationscay ybay andhay orfay ASANay.

Sentence 3: Ethay ENIACway adhay onay instructionsway, osay ethay omenwat adhay otay ogay roughthay ethay ardwarehay otay earnlay owhay otay usewat itway.

Rules Brainstorm:

NAME: ______________________

DATE: ______________________

5

CODE BREAKING WORKSHEET

PART ONE: The Caesar Cipher

A Caesar Cipher works by rotating the alphabet by a number of rotations. The example below shows one rotation. The **bottom** row is the original word and the **top** row is the **cipher**. So, A becomes B, B becomes C, C becomes D, and so on.

EXAMPLE USING ROT1:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Use the example cipher to decode the following word: **IJ! ____!**

Click for Caesar Cipher Template

Instructions:

1. As a class, encode the word ENIAC.

2. In your group, **encode** three 5-letter words using the worksheet.

- One of the words must decode to **ENIAC**.

3. You must use the same rotation for all three words.

4. Give your three ciphers to another group in a random order.

5. In your group, **decode** the three ciphers from another group.

NAME: ______________________

DATE: ______________________

CODE BREAKING WORKSHEET

PART TWO: The Extended Caesar Cipher

Instructions:

1. In your group, **encode** 3-5 long sentences using the worksheet. Consider avoiding words that might make your message easy to decode.
    - You **must** use the same rotation for all words.
2. Brainstorm strategies for how you'll decode another group's message. Consider how you might decode your own message.
3. Give your cipher to another group.
4. In your group, **decode** the cipher from another group.

Fun Fact!!

The Caesar Cipher is named after Julius Caesar because he used it for his private messages. It was in use for hundreds of years. A method to break it was first documented in the 9th Century called: Frequency Analysis. Al-Kindi, a mathematician, found that certain letters appeared more often in the Arabic by counting the letters in the Qur'an. He realized that by knowing which letters appear most often, he could guess the rotation used in a cipher and decode any message.

How does it work? Each language has its own order for which letters appear most often. This is known as that language's fingerprint. In English, E appears most often. If we're given a cipher, we start by finding the letter that appears most often. Let's pretend that F appears most often. Next, we rotate our wheel until E becomes F, which is the above ROT1. Now, we can decode the rest of the message using this rotation!

NAME: ______________________

DATE: ______________________

CODE BREAKING WORKSHEET

PART ONE: The Caesar Cipher

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z

ROTATION USED: __________ (between 1 and 25)			
ORIGINAL	**E N I A C**	_ _ _ _ _	_ _ _ _ _

PART TWO: The Extended Caesar Cipher

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z

ROTATION USED: __________ (between 1 and 25)
ORIGINAL

NAME: ______________________

DATE: ______________________

CODE BREAKING WORKSHEET

STRATEGIES BRAINSTORM

NAME: ____________________

DATE: ____________________

CODE BREAKING WORKSHEET

PART ONE: The Caesar Cipher

_ _ _ _ _	_ _ _ _ _	_ _ _ _ _	_ _ _ _ _

PART TWO: The Extended Caesar Cipher

NAME: ______________________

DATE: ______________________

CODE BREAKING WORKSHEET ANSWER KEY

PIG LATIN ACTIVITY:

Translations:

Sentence 1: There were no screens on the ENIAC, so the women used paper to keep track of everything.

Sentence 2: The first "computers" were women who did math calculations by hand for NASA.

Sentence 3: The ENIAC had no instructions, so the women had to go through the hardware to learn how to use it.

Rules:

If a word starts with:

- One letter consonant sound:
  - Move the consonant letter to the end of the word.
  - Add "ay" to the end.
- A two letter consonant sound:
  - Move both consonant letters to the end of the word.
  - Add "ay" to the end.
- Vowel sound:
  - Add "way" or "yay" (or just "ay") to the end of the word.

NAME: ______________________________

DATE: ______________________________

CODE BREAKING WORKSHEET ANSWER KEY

CODE BREAKING ACTIVITY (SAMPLE ANSWERS):

ROTATION USED: 1 (between 1 and 25)			
ORIGINAL	E N I A C	CODED	F O J B D

ACTIVITY 1 WITH ROT2 EXAMPLE:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X

ROTATION USED: 2 (between 1 and 25)			
ORIGINAL	E N I A C	H E L L O	W O R L D
CODED	G P K C E	J G N N Q	Y Q T N F

ABSTRACTION

HOW DO I ALREADY THINK LIKE A COMPUTER SCIENTIST?

FEATURING: **Brenda Wilkerson**

Description: Students determine key features for various audiences, identify shared qualities of multiple objects, and visually represent abstracted information.

LEARNING GOALS

Students will be able to...
1. Define "abstraction."
2. Perform abstraction.
3. Understand the importance of abstraction both in CS and elsewhere.

MATERIALS

1. Video: Computer Science Concept - Abstraction
2. Video: Empowering
3. Video: Typical Day
4. Worksheet

STANDARDS

CSTA: 1A-AP-09
CCSS ELA: CCRA.R.2
CCSS Math: MP7

VOCABULARY

Abstraction: The process of sharing or modeling only the most important parts of something

HOOK (10 minutes)

1. Share today's learning goals with the students.
2. Split students into groups of 4-5, and tell them that they are tasked with planning a party for their school. Clarify that this is a hypothetical party!
3. **Set a timer for 5 minutes**. Tell students to plan out what type of information needs to be given to the organizing committee and the guests.
   - **Suggested Examples:**
     - Caterer: Number of guests, dietary restrictions, budget, venue location
     - Decorator: Dress code, budget, color scheme, venue location
     - Guests: Dress code, menu, venue location
4. **Explain**: Share with the class that they have successfully abstracted an event by figuring out what each group needs to know and only sharing that specific information. Everything else that is irrelevant to that group was hidden, just like with abstraction!
5. Play the Computer Science Concept - Abstraction Video.
   - **Unplugged Alternative**: Have students read the "What is Abstraction?" section of the worksheet.
6. **Ask**: How did you use principles in this video in your event planning exercise? Have 1-2 students to share out their answers.

WOMAN IN TECH SPOTLIGHT (5 minutes)

1. Play the Brenda Wilkerson videos: Empowering and Typical Day
    - **Unplugged Alternative**: Read her biography on the worksheet.
1. **Explain**: The first video shared what Brenda believes girls and other minority groups are capable of accomplishing, and the second showed how she makes change through engaging with very different groups.
1. **Ask**: Who does Brenda Wilkerson interact with? How might Brenda Wilkerson use abstraction to communicate with many different types of people?
    - **Suggested Answers**: Brenda needed to get many different people on board with her program, and they all need different information based on their unique positions. Presenting information to someone in a way that they understand, relate to, and doesn't overwhelm them is an important part of change-making.

ACTIVITY (25 minutes)

1. Split the class into groups of 2-3 students, and assign numbers to each group.
2. **Explain**: Tell students that they are going to use their creative thinking and abstraction skills to model overlapping qualities in animals. Tell odd numbered groups that they will have animal group 1, and even numbered groups that they will have animal group 2.
3. **Set a timer for 7 minutes**. In their groups, students will complete the first page of the worksheet, where they will brainstorm qualities that the animals in their group share, and draw a new animal that has those features. After this, they can answer the reflection question.
4. **As a class**: Have students come together and share what their abstracted version looks like with the whole group. Have group 1 go first, while group 2 shares similarities and differences between them. Then switch groups and allow group 1 to comment on group 2's models.
5. **Highlight**: Different information leads to a different model, but the same information can also lead to different models based on what each group chooses to highlight and represent.
6. **Set a timer for 7 minutes**. In the same groups, ask students to move onto the second page of the worksheet when a koala bear is added into the model.
7. **As a class**: Have students come together and share what their final model looks like with the whole group. Note similarities and differences between the various models of the three animals. What changed when including a third animal in the model?
8. **Ask**:
    - What did you notice in these abstractions? Were they similar? Is this what you expected, and why?
    - Was it difficult to add the third animal in? How did you integrate the koala bear into your model? Did you start over with all three animals, or begin with the model you created in part one and add the third one?

DEBRIEF (5 minutes)

1. **Explain**: At first, you planned an event by giving different groups of people relevant, unique information based on their role in the event. Second, you took key features from multiple animals to create high-level representations of them.
2. **Ask**: How do these two examples relate to the idea of abstraction?
    - **Suggested Answers:**
        - The goal of abstraction is the cover up the unimportant information while highlighting the relevant details.
        - Once you've planned your event, it is fine for the guests to not know the budget of the event because it is not relevant.
        - Similarly, when you have your final model of the animals, it does not matter what the original animals were that you are modeling -- just the final representation that shares the most important, overlapping features.
3. **Reflect**: Think about examples in your life, academic or personal, where you could use modeling or abstraction. What does that look like? How often do you see that happening?
4. **Close out**: Note that by using abstraction, students are already thinking like computer scientists. Share that abstraction is a tool that is used by all computer scientists, and one that they can continue using in and out of the classroom.

WHAT IS ABSTRACTION?

Abstraction is the process of representing the most important parts of something without including complex background information or details. Computer scientists and engineers use abstraction to remove details from their software. This makes it easier for others to interpret its function.

John V. Guttag writes, "The essence of abstractions is preserving information that is relevant in a given context, and forgetting information that is irrelevant in that context."

Abstraction can be used in many situations. We only include important information when presenting a clear argument, and look for important details when reading long passages. Isolating the most relevant information is an important exercise in modeling. Computer scientists regularly model through abstraction.

BRENDA WILKERSON

Brenda Wilkerson is dedicated to enabling all students to access computer science, particularly students from underrepresented communities. After working in tech for several years, Brenda transitioned to working in computer science education. This is because she believes that "computer Science education...maximizes the innate potential of every student to become upstanding citizens, committed to lifelong learning, able to transform and advance their community, nation, and the world."

Brenda launched the "Computer Science for All" movement as the Director of Computer Science and IT Education for Chicago Public Schools. This initiative established formal requirements for CS education. In 2016, the Obama Administration decided to expand the project to the nation.

Now, as the Director of AnitaB.org, Brenda demonstrates the many ways people can contribute to tech. She focuses on how diverse groups of people can access computer science. Through her role, she inspires these groups to change the world through computer science.

How did Brenda communicate with different groups throughout her career? How could share have used abstraction in that process?

NAME: ______________________________

DATE: ______________________________

ABSTRACTION

INSTRUCTIONS: Follow your teacher's instructions to figure out what group of animals you will be abstracting below. Take your group's two animals and think about what qualities they share. If you were to abstract this to model the overlapping, most important features of these animals, what would that look like? Draw your abstracted model in the box below.

Group 1: Dog, Lion
Group 2: Penguin, Bear

What physical qualities come to mind when thinking about:

Animal 1 ______________________________ **?**

____

Animal 2 ______________________________ **?**

____

Are there overlapping physical qualities between these two animals that represent both of their features? What are these qualities?

- Example: 4 legs
- ______________________________
- ______________________________
- ______________________________

NAME: ______________________________

DATE: ______________________________

ABSTRACTION

A model that represents both of these animals is:

Would your model still apply to other groups of animals? Why or why not?

NAME: ______________________________

DATE: ______________________________

ABSTRACTION

INSTRUCTIONS: Now that you've completed one level of abstraction, we are going to add another layer to consider. Now, your model is going to include a third animal: **koala bear.**

What physical qualities come to mind when thinking about a **koala bear**?

____

What are the overlapping physical qualities between lions, dogs, and koala bears that you could highlight to model these animals in a way that works for all three of them? What are these qualities?

- Example: 2 eyes
- ______________________________
- ______________________________
- ______________________________

A model that represents both of these animals is:

How did you integrate the koala bear into your model?

USER TESTING

HOW IS USER TESTING SHAPING OUR WORLD?

FEATURING: **Chieko Asakawa**

Description: Students learn about user testing, design an activity for a particular user persona and practice giving and receiving feedback with their peers.

LEARNING GOALS

Students will be able to...

1. Explain the Identify the needs of a specific user
2. Modify a game to meet a user's needs.
3. Give specific, concrete feedback based on the needs of a user.

MATERIALS

1. Worksheet
2. Video: TED Talk

STANDARDS

CSTA: 1B-IC-19
CCSS ELA: CCRA.SL.1
CCSS Math: MP5

VOCABULARY

Prototype: An early sample or model of something that is made to learn about or test a concept or design
User experience: The feelings and attitudes a person has when they are using a product.
User persona: A made-up profile for a person who matches the characteristics of someone who might use a particular product.

HOOK (10 minutes)

1. Share today's learning goals with the students.
2. **Explain**: Today, you are going to create or modify a game to fit the needs of a specific audience. Your group will receive user personas based on chance.
3. Divide students into groups of 3-4, and give each of them dice.
4. Have students roll the dice three times and record their answers on the worksheet.
5. **Assign** user personas based on the numbers that students rolled on the dice..
    - First roll: Groups with even numbers will design for people over the age of 60, and those with odd numbers will design for elementary school students.
    - Second roll: Groups with even numbers will design for people who have limited sight and those with odd numbers will design for people who have limited hearing.
    - Third roll: Groups with even numbers will design something that is for competitive people and groups with odd numbers will design a game that does not have a winner for less competitive people.

6. **Set a timer for 2 minutes**: Have students complete the first part of the worksheet that says "Include" and "Avoid", where they brainstorm what they should or shouldn't have in their game based on their user persona.
    - **Example**: If we are designing for people over the age of 60, we may not want to incorporate too much movement. If we are designing for elementary schoolers, we may not want them to sit still for too long.
7. **Share**: Have each group share out some of their observations about what they should include and avoid based on their user persona.
8. **Set a timer for 5 minutes**: Ask students to create new game or modify an existing game. Each game should have three rules, which students should write down on the worksheet. They should match the rules of the game to their user persona

WOMAN IN TECH SPOTLIGHT (5 minutes)

1. Play the Chieko Asakawa TED Talk [3:20 - 4:40]
    - **Unplugged Alternative**: Her biography on the worksheet.
2. **Explain**: This video shows Chieko Asakawa's journey of designing technology that works for blind people, so that they can access the internet, too!
3. **Ask**: How might Chieko Asakawa conduct user testing to make sure that her technologies work for blind people?
    - **Suggested Answer**: She can share her prototypes with blind users to make sure that they are able to use the technology well.

ACTIVITY (25 minutes)

Part 1: Internal Test (10 min)

1. **Explain**: An Internal Test means that you are testing your game within your own group. This is so that you can see what works and what needs to be changed before sharing it with others.
2. **Set timer for 4 minutes**. Have students play their games within their groups. Remind the class to keep their user persona in mind and think about how that group would play the game.
3. Instruct students to discuss what worked well and what didn't and take notes on the chart on the worksheet. Some questions to consider:
    - Are your rules clear enough for other groups to understand?
    - Do you think your game meets the needs of your user persona?
    - Does your game match the interests of your user persona?
    - Are your instructions clear enough that your user persona can play and enjoy this game?
4. Have students make any changes before moving to Part 2.

ACTIVITY CONTINUED (25 minutes)

Part 2: Ground Swap (15 min)

1. **Explain**: Now you are going to conduct a Group Swap test. For this user test, you are going to swap your game with another group, One group will watch the other group play their game, then you will swap roles. When you are the players, you are going to take the rules and the user persona of the group you are swapping with, and play the game while considering the perspective of that persona.
2. **Set timer for 5 minutes**: One group plays the game while the other group watches and takes notes on what they are observing from the user test in the worksheet.
3. **Feedback**: Have the group that was playing the game provide feedback to the group that designed the game. The group that designed the game can also ask questions from the group who was testing the game.
4. **Set timer for 5 minutes**: Swap roles! Now the other group should play the game while the team who made the rules observes.
5. **Feedback**: Have the group that was playing the game provide feedback to the group that designed the game. The group that designed the game can also ask questions from the group who was testing the game.
6. **Ask**: Now, you have played your own game, modified it, and you have seen another group play your game. After observing someone else play your game and hearing their feedback, are there more changes you think you should make?

DEBRIEF (5 minutes)

1. **Explain:** Now you have practiced two different methods of user testing, and been able to change your game based on feedback. That is exactly what computer scientists do when building games, programs, and writing code.
2. **Discuss**:
    - Internal Test and Group Swap have different benefits. Which one felt more useful for this activity, and why?
    - Did you want to modify your game after both of these rounds? Why or why not?
    - Do you feel like your game matches your user persona? How can you tell?
    - How do you think your results would change if you were user testing with people who are actually fit your user persona? Why?

WHAT IS USER TESTING?

User testing is a way to check how people feel about specific ideas or products. It can be as simple as observing how people use a product, or as complicated as asking people to take a survey or give specific feedback about an idea. The goal is to collect information and make observations about what could make users' experiences better.

WHAT CAN USER TESTING HELP ACCOMPLISH?

User testing is a useful tool because it helps people understand someone else's perspective. Companies often use user testing and feedback to make improvements to their own product or service. This is particularly helpful when creating something with a specific goal or for a specific audience. For example, if you are building an app for your Spanish class final project, you could utilize user testing to make sure that it actually works for individuals who speak Spanish. Based on their feedback, you can change your design to be more helpful for them.

CHIEKO ASAKAWA

Chieko Asakawa is a blind Japanese computer scientist who transformed the way that people with disabilities have access to the internet, technology, and computer science. After getting her bachelor's degree in English Literature, she joined a course that taught programming to blind people. This course used an Optacon, which translates written language into things that she could feel and touch.

After this course, she became a researcher at IBM, where she built the IBM Home Page Reader. This tool is the most widely used web-to-speech system available. Web-to-speech systems make it possible to read any text on the web aloud. She also built a system that mimics they way people who are visually impaired experience the internet. This tool not only builds empathy, but it also allows for programmers to understand the challenges that visually impaired people may have when accessing the products or sites that they are designing.

Cheiko went on to receive her Ph.D In Engineering from the University of Tokyo, was added to the Women in Technology International Hall of Fame, and received the Medal of Honor with Purple Ribbon from the Japanese government for her work and impact on the field of technology.

How might Chieko Asakawa conduct user testing to make sure that her technologies work for people with visual impairments?

NAME: ______________________________

DATE: ______________________________

USER TESTING: CREATE YOUR GAME

INSTRUCTIONS: Fill out the following table to design your game based on your unique user persona.

DICE ROLL #1:

DICE ROLL #2:

DICE ROLL #3:

Therefore, my group is designing a game for someone who is ______________________________,

______________________________, and ____.

Use the space below to brainstorm what has to be included in your game based on your user persona, and what you should avoid?

INCLUDE	*AVOID*

4

NAME: ______________________________

DATE: ______________________________

USER TESTING: CREATE YOUR GAME

Based on your brainstorm, what are the three rules for your game?

1.

2.

3.

NAME: ______________________________

DATE: ______________________________

USER TESTING: CREATE YOUR GAME

Part 1: Internal Test

INSTRUCTIONS: Use the space below to document things that you like about your game and want to keep, and also things that you want to change based on your internal test. Think about the following questions:

- What could make this game more fun?
- Are all three of your instructions helpful and important to the game?
- Does this game fit your user persona? If not, what could you change to make it fit the persona more?

KEEP	*CHANGE*

Based on your reflection, what are the three rules for your game? If they are the same, write the same rules again?

1.

2.

3.

NAME: ______________________________

DATE: ______________________________

USER TESTING: CREATE YOUR GAME

Part 2: Group Swap

INSTRUCTIONS: Use the space below to write down observations about your game when you are watching the other group play it. Also write down any questions you may want to ask the group after they have played the game.

Note: Remember your specific user persona when writing your observations and questions, because the game is for them!

OBSERVATIONS	*QUESTIONS*

When the group is giving you feedback about your game, take notes here. You can also write down their answers to your questions that you wrote above.
