Title: “iGrasshopper”	Targeted Grade: 6-12 Lexile: 610L-800L
Author(s):	Time Expectancy: ~90 min Depth of Knowledge (DOK 1, 2, or 3): 2-3
Computer Science Learning Objectives: Student will: ● create and customize an original sprite using drawing tools and explore the concept of grouping and ungrouping sprites within the Scratch programming environment. ● apply coding concepts to control the movement and appearance of their sprite by utilizing code blocks for changing colors, adding sounds, and implementing repeat loops.	
Concepts/Keywords: Scratch, Sprite, grouping, loops, blocks, coding, coordinates (X and Y), coordinate plane, axis(es), sequence, debugging	
K-12 CSTA Identifier(s)	**Standard(s)** *and Descriptive Statement(s)*
2-AP-10	**-Use flowcharts and/or pseudocode to address complex problems as algorithms.** (Subconcept: Algorithms; Practice: 4.4, 4.1)
2-AP-11	**-Create clearly named variables that represent different data types and perform operations on their values.** (Subconcept: Variables; Practice: 5.1, 5.2)
2-AP-12	**-Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.** (Subconcept: Control; Practice: 5.1, 5.2)
1A-AP-14	-**Debug (identify and fix) errors in an algorithm or program that includes sequences and simple loops.** (Subconcept: Program Development; Practice: 6.2)
1A-IC-18	**-Keep login information private, and log off of devices appropriately.** (Subconcepts: Safety Law & Ethics; Practice: 7.3)
K-12 Computer Framework(s)	Practice # and Statement(s)
P1. Fostering an Inclusive Computing Culture	1. **Include the unique perspective of others** and reflect on one’s own perspectives when designing and developing computational products.
P4. Developing and Using Abstractions	1. **Extract common features** from a set of interrelated processes or phenomena. 2. **Evaluate existing technological functionalities** and **incorporate them** into new designs.

	3. **Create modules** and **develop points of interaction** that can (continue from previous page) apply to multiple situations and reduce complexity.
P5. Creating Computational Artifacts	2. **Create a computational artifact** for practical intent, expression, or to address a societal issue.
ISTE Standards	Standard(s)/Statement(s)
2. Digital Citizen: Students recognize the rights, responsibilities and opportunities of living, learning and working in an interconnected digital world, and they act and model in ways that are safe, legal and ethical.	2d. Students manage their personal data to maintain digital privacy and security and are aware of data-collection technology used to track their navigation online.
4. Innovative Designer: Students use a variety of technologies within a design process to identify and solve problems by creating new, useful or imaginative solutions.	4c. Students develop, test and refine prototypes as part of a cyclical design process. 4d. Students exhibit a tolerance for ambiguity, perseverance and the capacity to work with open-ended problems.
5. Computational Thinker: Students develop and employ strategies for understanding and solving problems in ways that leverage the power of technological methods to develop and test solutions.	5d. Students understand how automation works and use algorithmic thinking to develop a sequence of steps to create and test automated solutions.
Additional Content Standard #(s)	Standard(s)/Statement(s)
<u>NGSS</u>: MS-ETS1-3	Analyze data from tests to determine similarities and differences among several design solutions to identify the best characteristics of each that can be combined into a new solution to better meet the criteria for success. **SEP**:Developing Models **DCI**: ETS1.C: Optimizing the Design Solution
MS-ETS1-4	Develop a model to generate data for iterative testing and modification of a proposed object, tool, or process such that an optimal design can be achieved. **SEP**: Developing and Using Models **DCI**: ETS1.B: Developing Possible Solutions
<u>CCSS-ELA</u>: RI.7.3 Key Ideals and Details	Analyze the interactions between individuals, events, and ideas in a text (e.g., how ideas influence individuals or events, or how individuals influence ideas or events).

RI.7.4 Craft and Structure	Determine the meaning of words and phrases as they are used in a text, including figurative, connotative, and technical meanings; (contined from previous page) analyze the impact of a specific word choice on meaning and tone.
State (or International) Standard(s): (TBD and identified by location of instructor utilizing lesson).	
References	K-12 CSTA Standards: Computer Science Teachers Association (2017). *CSTA K–12 Computer Science Standards, Revised 2017.* Retrieved from https://csteachers.org/k12standards/. K-12 Computer Science Framework: https://k12cs.org/wp-content/uploads/2016/09/K%E2%80%9312-Computer-Science-Framework.pdf Next Generation Science Standards: https://www.nextgenscience.org/standards/standards Common Core State Standards for ELA: http://www.thecorestandards.org/ELA-Literacy/ ISTE Standards: https://www.iste.org/standards/for-students Bloom's Digital Taxonomy Verbs: https://libguides.bc.edu/c.php?g=628962&p=4506921 Scratch: https://scratch.mit.edu/ Engineer Girl: https://www.engineergirl.org/123598/Grace-Hopper Computer History: https://www.computerhistory.org/timeline/
Lesson Resources/Folder Access (Link)	

Overview:
The objective of the "iGRASSHOPPER" lesson is to guide students through the process of creating their own sprite, controlling its movements using code blocks, and exploring various features of the Scratch interface. Students will learn how to manipulate sprites, change their appearance, and make them move using simple commands. The lesson culminates with an extension activity that encourages students to explore the history of computer development and its connection to the present.

In Part I of this lesson, students will learn how to delete the default sprite and create their own by utilizing the drawing tools and experimenting with the "Fill" and "Outline" features. The goal is to create an original "grasshopper robot" sprite. Students will take a screenshot of their creation to document their progress.

In Part II, students will explore the various functionalities available in the Scratch interface for manipulating their sprites. They will experiment with commands like "Flip Horizontal," "Flip Vertical," "Copy," "Paste," and "Group" to understand their effects on sprite manipulation. The concept of grouping and ungrouping sprites will be introduced to students. They will also learn to move and position their sprites using both manual interaction with the X and Y coordinates and keyboard commands.

In Part III, students will delve into using code blocks to control the movements of their grasshopper sprite. They will learn how to change the color and appearance of the sprite, add sounds, and create repetitive actions using the "repeat control" block. Students will experiment with coding sequences to achieve desired effects and explore the relationship between code blocks and sprite behavior. At this point, students will be asked how they can make the code run forever and will be introduced to forever loops.

Finally the Extend Your Thinking section of the lesson explores the life and contributions of Grace Hopper through provided resources. Students are encouraged to reflect on the broader impact of computing and its historical context. This extension challenges students to think critically about the societal implications of technological advancements and the role of innovative thinkers like Grace Hopper.

Rationale/Background: This lesson is designed to engage students in hands-on programming activities while introducing them to concepts of coding, such as creating, manipulating, and controlling sprites, a variable. The culmination of the lesson introduces students to the especially fundamental coding concept of loops. The Scratch environment provides an accessible platform for students to develop their computational thinking skills and creativity. By encouraging experimentation and exploration, this lesson fosters a deeper understanding of coding principles and empowers students to express themselves through digital creations.

The students participating in this lesson are assumed to have basic computer literacy and familiarity with web browsing. The lesson's activities cater to a diverse range of learners by providing step-by-step instructions and opportunities for creative expression. The extension activity involving the exploration of Grace Hopper's contributions to computer science adds historical context and encourages critical thinking about the evolution of technology.

Teacher (Required) Materials/Resources: Hard copy or digital copy of this lesson for the instructor and each student (if teaching this lesson in person), Internet access, a means of meeting virtually (either by Google Meet, Zoom [although Zoom is not preferred due to threat of hacking and inappropriate disruptions], or FaceTime (depending on the size of the intended class).

“iGRASSHOPPER Appendix 1”- Internet-Free Supplement: This document is provided in the event that any student does not have access to the Internet in their home environment and the “Extend Your Thinking” portion is assigned to be completed as homework.

Student Materials:

1. An iPad, laptop or desktop computer.
2. Internet access.
3. Either hard copy or digital copy (that will allow writing on [such as Notability]) of “Student Sheet: iGrasshopper”.
4. Scratch accounts created during the previous lesson.
5. Ability to screenshot or photograph their assembled block sequences for each tutorial category.
    a. Encourage students to look up how to screenshot on their own devices, and assist when necessary.
    b. Example: Use the Google search engine to search “how to screenshot on Windows 10”

6. For “Extend Your Thinking: Admiral of Algorithms”: Either hard copy or digital copy of reading material. Students may follow directions to the link provided on their student sheet, or print a hardcopy from “iGRASSHOPPER Appendix 1.”

Guided Practice/Instructor Procedures:

A) Introduction and Motivation:
   1. Begin the lesson by reviewing the previous two session's content where students learned how to give detailed instructions to a computer and explored the Scratch interface.
   2. Introduce the concept of creating and customizing sprites to express individual creativity.
   3. State the lesson’s purpose: To create a custom spite and continue exploring block types with it, especially the “loop” and “forever loop.”

B) Lesson Body:

Part I: Creating Original Sprites

   1. Guide students through the process of accessing the Scratch interface and deleting the default sprite.
   2. If needed, demonstrate how to create a "grasshopper robot" sprite using drawing tools, fill, and outline features.
   3. Instruct students to create their own grasshopper sprite using the provided reference image on their student sheet.
   4. Instruct students to take a screenshot of their sprite and place it in the provided box.

Part II: Making It Move

   1. Instruct students to complete the prompts and answer the questions in Part II of their student sheets.
   2. Allow students time to explore the available buttons as desired.
   3. Facilitate a class discussion about the effects of grouping and ungrouping sprites.
      a. “What happens when you group two or more sprites together? How does this affect their behavior?”
      b. “Can you think of a scenario where you might want to group sprites? How would grouping help in that situation?”
      c. “How does ungrouping sprites impact their individual behavior? Why might you choose to ungroup them?”
      d. "Imagine you're designing a game with multiple characters, objects, and background elements. How might grouping them together make it easier to manipulate and control their behaviors as a whole?"
      e. "Consider a scenario where you're designing an animation with several moving parts. How could ungrouping certain elements allow you to fine-tune their individual movements?"
   4. Facilitate a class discussion on the importance of coordinates.
      a. “What do the X and Y coordinates represent when working with sprites on a canvas?”
      b. “Imagine you're creating a game where a sprite needs to move to specific locations. How would you use coordinates to achieve this movement?”

c. “Think about a website you visit frequently. How are different elements, like images and text, positioned on the webpage? How might understanding coordinates be helpful in arranging these elements?”
d. “Can you think of a real-world analogy where understanding coordinates is crucial for precise positioning or navigation?”
    i. Maps or other location devices, web-design, app design, etc.
    ii. Answer may not be immediately related to coding, but encourage students to relate it back, especially higher-level learners. For example, paper or flat maps locations can be described using coordinates, which can then be incorporated into code for a location-related cell phone notification.

5. Assure students have completed the last prompt to delete one sprite and center the remaining sprite.

Part III: Using Code To Control Movements

1. Introduce students to code blocks and their role in controlling sprite behavior.
2. Again instruct students to follow the prompts and answer the questions in Part III of their student sheets.
3. Encourage students to move beyond the provided prompts if desired to play with what they can make their grasshopper do.
    a. Higher-level students may be inspired by encouragement to create a story with their sprites movements, think of how this sprite could be used in a video game, or pretend it is giving a theater performance.
4. As students will inevitably run into problems trying to get the sprite to perform as desired, sacilitate a discussion about debugging and its importance in coding.
5. Culminate the lesson by facilitating discussion about the last prompt which introduces loops, an integral part of coding.
    a. “Why would loops be important in coding?”
    b. “How might loops contribute to writing cleaner and more organized code?”
    c. “If you had to change the color of a sprite's costume multiple times, how could using a loop make the process more efficient?”
    d. "Think about our first lesson together. How could we use a 'loop' in our previous instructions to make our sandwiches or brush our teeth?"
    e. “Why would you choose a repeat loop (limited) or a forever loop?”
    f. “Can you think of a situation where a loop could potentially lead to unintended consequences or an ‘infinite loop’? How would you prevent or fix that?”
        i. Note intentional vs unintentional looping, and encourage students to think critically about when NOT to use a loop.
        ii. You may also be able to introduce “breaking” a loop here.

C) <u>Lesson Closure:</u>

1. Ask students to share their experiences and any challenges they faced during the lesson.
2. Ask the students to summarize the key concepts covered in the lesson such as Sprites, sequences, loops, and debugging.
3. Extend Your Thinking reading can be completed during the end of class if time allows, or be assigned as homework. Discuss in class if possible.

Student Misconceptions:

- Grouping and ungrouping only affect the visual arrangement of sprites.
    - Clarifying the impact on code execution is crucial.
- Code will or should work perfectly on the first attempt, or computer can understand and fix errors in code.
    - Coding often involves experimentation, refinement, and continuous improvement.
    - Express the importance of debugging.
- There is only one correct way to achieve a certain outcome in coding.
    - Coding requires creativity and problem-solving, allowing multiple paths to reach a desire result.

Reading Selection: Direct students to the provided link or printed “iGRASSHOPPER Appendix 1” to read about Grace Hopper's contributions to computer science. Instruct them to watch the embedded video and answer the comprehension questions based on the information provided in the article and video. Encourage students to explore the Computer History Museum's timeline of computer history to gain insight into the evolution of technology.

Assessment:

A) Student assessment (by instructor):

Informal Assessment: During the lesson, the instructor will conduct ongoing formative assessments by circulating among students as they work on their sprite creations and code blocks. The instructor observe their engagement, problem-solving approaches, and interactions with the Scratch interface. The instructor can ask open-ended questions to gauge their understanding of concepts, such as their rationale for selecting specific commands or their strategies for making the sprite move. Lastly, the instructor will review students' screenshots of their grasshopper robot creations to assess their ability to apply drawing tools and create an original sprite.

Formal Assessment: The instructor will assess students' comprehension of computational concepts and practices through written reflections or discussions about the role of code blocks, the importance of debugging, and their insights from exploring the Extend Your Thinking section on Grace Hopper.

B) Instructor Self and Student Evaluation: The instructor is encouraged to complete the following as the lesson is being carried out or reflected after the lesson is completed.

Three Strengths of This Lesson:

1) _____
2) _____
3) _____

Three Elements/Areas for Improvement:

1) _____
2) _____
3) _____

Identification of students (**using initials, not names**) who were not successful in meeting the stated objectives: _____
_____

How shortcomings will be addressed prior to starting next session:
_____
_____
_____

Scope and Sequence: Prior Assignment: “SCRATCHing the Surface”
Next Lesson: “HOPportunities”

Look-Ahead: In the next lesson, "HOPportunities," students will again need access to the Internet and their Scratch accounts. They will explore more sprite animation and interactions using complex movements, timing, and learn to create interactive stories.