Title: Jitterbugs	Targeted Grade: 6-12 Lexile: 810L - 1400L
Author(s):	Time Expectancy: ~90 minutes Depth of Knowledge (DOK 1, 2, or 3): 3
Computer Science Learning Objectives: Student will: ● apply coding principles to tell narrative through Scratch, creating and animating sprites that correspond to characters from the story. ● create a program of dancing Sprites and utilize sound blocks. ● explore historical contexts and cultural significance through the lens of the Jitterbug dance and computing pioneers.	
Concepts/Keywords: Scratch, storytelling, animation, sprites, code block, program, sequence, loop, historical context	
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-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. 3. **Create modules** and **develop points of interaction** that can apply to multiple situations and reduce complexity. 4. **Model phenomena and processes** and **simulate systems** to understand and evaluate potential outcomes.

P5. Creating Computational Artifacts	2. **Create a computational artifact** for practical intent, expression, or to address a societal issue.
P6. Testing and Refining Computational Artifacts	1. **Systematically test** computational artifacts by considering all scenarios and using test cases. 2. **Identify and fix** errors using a systematic process. 3. **Evaluate and refine** a computational artifact multiple times to enhance its performance, reliability, usability, and accessibility.
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.	4a. Students know and use a deliberate design process for generating ideas, testing theories, creating innovative artifacts or salving authentic problems. 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.	5c. Students break problems into component parts, extract key information, and develop descriptive models to understand complex systems or facilitate problem-solving. 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)
NGSS: 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
CCSS-ELA: RL.6.1 Key Ideas and Details	Cite textual evidence to support analysis of what the text says explicitly as well as inferences drawn from the text.
RI.6.2 Key Ideas and Details	Determine a central idea of a text and how it is conveyed through particular details; provide a summary of the text distinct from personal opinions or judgments.
RI.6.7 Key Ideas and Details	Integrate information presented in different media or formats (e.g., visually, quantitatively) as well as in words to develop a coherent understanding of a topic or issue.
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/ YouTube: https://www.youtube.com/watch?v=trqXu5es84U Free Music Archive: https://freemusicarchive.org/genre/Big_BandSwing
Lesson Resources/Folder Access (Link)	

Overview: The goal of the "Jitterbugs" lesson is to lead students through a journey involving storytelling, coding, and historical insights. This lesson encourages students to embrace creativity, practice coding skills, and explore the history of dance and computing pioneers.

In Part I, "Giving an Inch, Going a Mile," students are immersed in the story of Ivy and Oscar, two inchworm friends. This tale, set in The Vibrant Garden, highlights the power of friendship, courage, and support. Through Ivy and Oscar's adventure, students are introduced to the themes of empathy, determination, and the importance of companionship. Students then return to the Scratch interface to create sprites that depict Ivy and Oscar and animate the story, applying their previous knowledge of sprite manipulation and design. This engaging activity enables students to exercise their creative storytelling skills while reinforcing their coding capabilities.

Part II, "Jitterbugs," introduces Alex and Jordan, sibling ladybugs. Students delve into the history of the "Jitterbug" dance, formerly known as the lindy hop. This dance's origin and significance within its cultural context may sparke discussions on social dynamics, self-expression, and historical narratives. Students again return to the Scratch interface to design sprites for Alex and Jordan and choreograph them in a Jitterbug dance routine. This coding endeavor prompts students to apply their coding skills to capture movement and rhythm, blending storytelling with technical execution.

The lesson culminates with an "Extend Your Thinking" segment that explores the life and contributions of Alan Turing. Students discover Turing's significant impact on mathematics, computer science, artificial intelligence, and more. This segment encourages reflection on the intersections between science, mathematics, and the advancement of technology.

By combining storytelling, coding, and historical context, the "Jitterbugs" lesson encourages students to develop well-rounded computational thinking, creativity, communication, and an appreciation for the interconnectedness of different disciplines. This holistic approach aims to inspire students to view computer science as a dynamic field with profound ties to various aspects of human culture and history.

Rationale/Background: The "Jitterbugs" lesson brings coding to life through imaginative narratives and historical context. By utilizing Scratch, students experience hands-on coding and animation techniques while enhancing their problem-solving skills. Delving into the origin of the Jitterbug dance familizaries students with social context, and Alan Turing's work fosters an appreciation for the evolution of technology and its impact. This lesson encourages creativity, collaboration, and critical thinking within a coding context.

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).

Student Materials:

1. An iPad, laptop or desktop computer, ideally with speakers for playing music.
2. Internet access.
3. Either hard copy or digital copy (that will allow writing on [such as Notability]) of “Student Sheet: Jitterbugs”.
4. Scratch accounts created during a 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”

Guided Practice/Instructor Procedures:

A) Introduction and Motivation

1. Engage students in a discussion to recap the previous lesson, “HOPportunitites.
    a. What do they recall enjoying most? What didn’t they like?
        i. Take notes and try to apply more of what was enjoyed and address or avoid dislikes where possible.
    b. Has anyone used Stratch to create something on their own?
2. Explain this lesson will continue to use storytelling to facilitate learning code, and encourage students to apply their own creativity throughout.

B) Lesson Body

Part I: Giving an Inch, Going a Mile

1. Read the short story on page one, titled "A Heart-Worming Friendship" aloud for the class. This story introduces the characters Ivy and Oscar, emphasizing the theme of friendship.
2. Facilitate a brief classroom discussion about the story. Encourage students to share their thoughts on the importance of friendship and how Ivy and Oscar's friendship develops. This discussion will help them connect with the story's themes.
3. Instructor students to access their Scratch accounts, and follow the prompts in the Student Sheet to create sprites and animate the story.
4. Assist students who are still moving slowly in the Scratch interface or still learning the code block concepts.
5. Encourage students who may quickly complete the assignment to add more details to their animation, add to the original story, or collaborate with other students.
6. Assure that students have pasted a screenshot of their Ivy and Oscar sprites on their Student Sheet before moving to the next section.

Part II: Jitterbugs

1. Again read aloud, or allow the students to read to themselves, the story on page 3 of their Student Sheets.
2. Briefly engage students in discussion about the lindy hop and jitterbug.
    a. “What were your initial thoughts about the Jitterbugs story and the history of the lindy hop/jitterbug dance?"
    b. “The lindy hop had significant cultural and social implications. How did this dance challenge the norms of its time, particularly regarding race and class?"
    c. “How did the lindy hop help individuals express themselves and escape societal constraints during its time?"
    d. “How did the popularity of the lindy hop evolve, and why do you think it eventually became known as the jitterbug?"
    e. “Do you believe that preserving and celebrating the history of dances like the lindy hop is important? Why or why not?"
3. Instruct students to follow the link in their Student Sheets to a video of the Jitterbug steps.
4. Inform the students they’ll now have the opportunity to preserve this historical dance through their coding and animation.
5. Instruct students to answers the question and follow the instructions in their Student Sheets.

6. Continue to assist students where they need support, pair students with complimenting strengths and weaknesses to support each other, and encourage excelling students to add details or extra story elements to their animations.
    a. Students may especially require support in downloading and uploading a song for their dance animation.
7. Assure that students have pasted screenshots of their code blocks on their Student Sheet, that you can access the project link, and initial their sheets.
8. Culminate the coding time by facilitating discussion about animating the two scenarios. Ask questions such as:
    a. “Which scenario did you enjoy animating more? Why?”
    b. “What were the key coding concepts and skills you applied when animating the "Heart-Worming Friendship" story and the Jitterbugs dance?”
    c. “Did you change how you used loops, grouping, or coordinates since the last lesson? Why or why not?”
    d. “Did you use any new coding concepts or blocks you hadn’t used yet?”
    e. “Explain the significance of incorporating music into the Jitterbugs dance animation. How did you integrate sound into your project, and what impact did it have on the overall experience?
    f. “What are some basic animations you've seen online that made you laugh or learn something new? What do you think made them effective? Did you or how could you apply those concepts to your own animations?”

C) Lesson Closure
1. Extend Your Thinking reading can be completed during the end of class if time allows, or be assigned as homework. Discuss question five, “Turing did research on morphogenesis and attempted to explain the presence of Fibonacci numbers in plant leaf patterns. Where else do you think there is a link between math and the natural world?” in class if possible.
2. Summarize the main concepts covered in the lesson:
    a. Reinforcing previously learned essentials like sequences, loops, dialogue, and sprite interactions.
    b. The real-world application of coding to tell stories that matter, like importance of friendship and historical events.

Student Misconceptions:
- As students continually apply the basic Scratch concepts in different ways, they may believe coding is about memorizing these concepts.
    - Encourage problem-solving and critical thinking instead of rote memorization. Show them how to find resources, seek help when needed, or even use their own previously created code/programs. Let them know that professionals Google all the time!
- Despite the storytelling, students may still think of coding as being limited to technical fields.
    - Coding can be applied to and in various domains, including art, science, healthcare, video games, sports and more. Ensure students that especially today, in almost every area of interest there is a way to utilize computer science and coding within that domain. In this lesson alone, we see that Alan Turning contributed to mathematics, international world events, human developmental science, and computer science.

Reading Selection: Direct students to the “Extend Your Thinking” section of the lesson starting on page seven of their Student Sheet. Instruct them answer the questions about alan Turning and his contributions to technological developments.

Assessment:

A) Student assessment (by instructor):

Informal Assessment:The instructor will engage in ongoing formative assessments by actively interacting with students while they work on designing their animations within the Scratch platform. The instructor will observe students' level of engagement, their problem-solving strategies, and their interaction with the Scratch interface. The instructor can ask open-ended questions to probe their understanding of key concepts, such as their rationale for choosing specific code blocks or their methods for animating the sprite's movements, as well as uses for storytelling and relevance of historical context. This real-time feedback will provide insights into students' grasp of the material and their ability to apply coding techniques effectively. Additionally, the instructor will review the animations to assess students' mastery of Scratch.

Formal Assessment: To formally assess students' comprehension of computational concepts and coding practices, the instructor will guide them in written reflections or class discussions. Students will be asked to elaborate on the role and significance of code blocks in programming, emphasizing how sequences of code contribute to sprite movements and interactions. This assessment will gauge their grasp of foundational coding principles and their ability to articulate these concepts effectively. Additionally, students will share insights from the content of the stories and context for each animation, demonstrating their understanding of storytelling as a communication tool and historical events. Through these assessments, the instructor can ascertain students' depth of understanding and critical thinking skills in the context of coding, communication, and history.

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: “HOPportunities”

Look-Ahead: *Provide a list of preparations to make for the next lesson. This can be written as a bulleted or numbered list, or written in paragraph form.*