

Discovery Education – Coding

Links to CAS pathways and K-12 Computer Science Framework

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
Block coding: Level 1			
On the Move: Under the Sea	To understand that when a computer does something, it is following instructions called 'code'.		
Links	<p><u>Programming and development</u> Knows that users can develop their own programs, and can demonstrate this by creating a simple program in an environment that does not rely on text. (AL) Executes, checks and changes programs. (AL) Understands that programs execute by following precise instructions. (AL)</p>	<p><u>K–2. Algorithms and Programming. Algorithms</u> People follow and create processes as part of daily life. Many of these processes can be expressed as algorithms that computers can follow.</p> <p><u>K–2. Algorithms and Programming. Variables</u> Information in the real world can be represented in computer programs. Programs store and manipulate data, such as numbers, words, colors, and images. The type of data determines the actions and attributes associated with it.</p> <p><u>K–2. Algorithms and Programming. Control</u> Computers follow precise sequences of instructions that automate tasks. Program execution can also be nonsequential by repeating patterns of instructions and using events to initiate instructions.</p> <p><u>K–2. Algorithms and Programming. Program Development</u> People develop programs collaboratively and for a purpose, such as expressing ideas or addressing problems.</p>	<p><u>Level 1A</u></p> <p><u>Algorithms and Programming</u> 1A-AP-08 Model daily processes by creating and following algorithms (sets of step-by-step instructions) to complete tasks.</p> <p>1A-AP-09 Model the way programs store and manipulate data by using numbers or other symbols to represent information.</p> <p>1A-AP-10 Develop programs with sequences and simple loops, to express ideas or address a problem.</p> <p>1A-AP-11 Decompose (break down) the steps needed to solve a problem into a precise sequence of instructions.</p> <p>1A-AP-12 Develop plans that describe a program’s sequence of events, goals, and expected outcomes.</p>
On the Move: Royal Chase	To give instructions to make objects on the screen move when the program starts.		
Links	<p><u>Programming and development</u> Knows that users can develop their own programs, and can demonstrate this by creating a simple program in an environment that does not rely on text. (AL) Executes, checks and changes programs. (AL) Understands that programs execute by following precise instructions. (AL) <u>Algorithms</u></p>	<p><u>K–2. Computing Systems. Devices</u> People use computing devices to perform a variety of tasks accurately and quickly. Computing devices interpret and follow the instructions they are given literally.</p> <p><u>K–2. Computing Systems. Troubleshooting</u> Computing systems might not work as expected because of hardware or software problems. Clearly describing a problem is the first step toward finding a solution.</p>	<p><u>Level 1A</u></p> <p><u>Algorithms and Programming</u> 1A-AP-08 Model daily processes by creating and following algorithms (sets of step-by-step instructions) to complete tasks.</p> <p>1A-AP-09 Model the way programs store and manipulate data by using numbers or other symbols to represent information.</p>

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
	<p>Understands what an algorithm is and is able to express simple linear (non-branching) algorithms symbolically. (AL)</p>	<p><u>K-2. Algorithms and Programming. Algorithms</u> People follow and create processes as part of daily life. Many of these processes can be expressed as algorithms that computers can follow.</p> <p><u>K-2. Algorithms and Programming. Variables</u> Information in the real world can be represented in computer programs. Programs store and manipulate data, such as numbers, words, colors, and images. The type of data determines the actions and attributes associated with it.</p> <p><u>K-2. Algorithms and Programming. Control</u> Computers follow precise sequences of instructions that automate tasks. Program execution can also be nonsequential by repeating patterns of instructions and using events to initiate instructions.</p> <p><u>K-2. Algorithms and Programming. Program Development</u> People develop programs collaboratively and for a purpose, such as expressing ideas or addressing problems.</p>	<p>1A-AP-10 Develop programs with sequences and simple loops, to express ideas or address a problem.</p> <p>1A-AP-11 Decompose (break down) the steps needed to solve a problem into a precise sequence of instructions.</p> <p>1A-AP-12 Develop plans that describe a program’s sequence of events, goals, and expected outcomes.</p>
<p>On the Move: Transport on the Go</p>	<p>To use code to make objects move when they are clicked on.</p>		
<p>Links</p>	<p><u>Programming and development</u> Knows that users can develop their own programs, and can demonstrate this by creating a simple program in an environment that does not rely on text. (AL) Executes, checks and changes programs. (AL) Understands that programs execute by following precise instructions. (AL) <u>Algorithms</u> Understands what an algorithm is and is able to express simple linear (non-branching) algorithms symbolically. (AL)</p>	<p><u>K-2. Computing Systems. Devices</u> People use computing devices to perform a variety of tasks accurately and quickly. Computing devices interpret and follow the instructions they are given literally.</p> <p><u>K-2. Computing Systems. Troubleshooting</u> Computing systems might not work as expected because of hardware or software problems. Clearly describing a problem is the first step toward finding a solution.</p> <p><u>K-2. Algorithms and Programming. Algorithms</u> People follow and create processes as part of daily life. Many of these processes can be expressed as algorithms that computers can follow.</p> <p><u>K-2. Algorithms and Programming. Variables</u> Information in the real world can be represented in computer programs. Programs store and manipulate data, such as numbers, words, colors, and images. The type of data determines the actions and attributes associated with it.</p> <p><u>K-2. Algorithms and Programming. Control</u> Computers follow precise sequences of instructions that automate tasks. Program execution can also be nonsequential by repeating patterns of instructions and using events to initiate instructions.</p> <p><u>K-2. Algorithms and Programming. Program Development</u> People develop programs collaboratively and for a purpose, such as expressing ideas or addressing problems.</p>	<p><u>Level 1A</u></p> <p><u>Algorithms and Programming</u> 1A-AP-08 Model daily processes by creating and following algorithms (sets of step-by-step instructions) to complete tasks.</p> <p>1A-AP-09 Model the way programs store and manipulate data by using numbers or other symbols to represent information.</p> <p>1A-AP-10 Develop programs with sequences and simple loops, to express ideas or address a problem.</p> <p>1A-AP-11 Decompose (break down) the steps needed to solve a problem into a precise sequence of instructions.</p> <p>1A-AP-12 Develop plans that describe a program’s sequence of events, goals, and expected outcomes.</p>

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
On the Move: Another Planet	To use code to write a computer program where objects in a space scene move when they are clicked on.		
Links	<p>Programming and development Knows that users can develop their own programs, and can demonstrate this by creating a simple program in an environment that does not rely on text. (AL) Executes, checks and changes programs. (AL) Understands that programs execute by following precise instructions. (AL)</p>	<p>K-2. Computing Systems. Devices People use computing devices to perform a variety of tasks accurately and quickly. Computing devices interpret and follow the instructions they are given literally.</p> <p>K-2. Computing Systems. Troubleshooting Computing systems might not work as expected because of hardware or software problems. Clearly describing a problem is the first step toward finding a solution.</p> <p>K-2. Algorithms and Programming. Algorithms People follow and create processes as part of daily life. Many of these processes can be expressed as algorithms that computers can follow.</p> <p>K-2. Algorithms and Programming. Variables Information in the real world can be represented in computer programs. Programs store and manipulate data, such as numbers, words, colors, and images. The type of data determines the actions and attributes associated with it.</p> <p>K-2. Algorithms and Programming. Control Computers follow precise sequences of instructions that automate tasks. Program execution can also be nonsequential by repeating patterns of instructions and using events to initiate instructions.</p> <p>K-2. Algorithms and Programming. Program Development People develop programs collaboratively and for a purpose, such as expressing ideas or addressing problems.</p>	<p>Level 1A</p> <p>Algorithms and Programming 1A-AP-08 Model daily processes by creating and following algorithms (sets of step-by-step instructions) to complete tasks.</p> <p>1A-AP-09 Model the way programs store and manipulate data by using numbers or other symbols to represent information.</p> <p>1A-AP-10 Develop programs with sequences and simple loops, to express ideas or address a problem.</p> <p>1A-AP-11 Decompose (break down) the steps needed to solve a problem into a precise sequence of instructions.</p> <p>1A-AP-12 Develop plans that describe a program’s sequence of events, goals, and expected outcomes.</p>
Simple inputs: Burst the bubbles	To combine start events and click events to make a simple game.		
Links	<p>Programming and development Knows that users can develop their own programs, and can demonstrate this by creating a simple program in an environment that does not rely on text. (AL) Executes, checks and changes programs. (AL) Understands that programs execute by following precise instructions. (AL)</p>	<p>K-2. Computing Systems. Devices People use computing devices to perform a variety of tasks accurately and quickly. Computing devices interpret and follow the instructions they are given literally.</p> <p>K-2. Computing Systems. Troubleshooting Computing systems might not work as expected because of hardware or software problems. Clearly describing a problem is the first step toward finding a solution.</p> <p>K-2. Algorithms and Programming. Algorithms People follow and create processes as part of daily life. Many of these processes can be expressed as algorithms that computers can follow.</p> <p>K-2. Algorithms and Programming. Variables</p>	<p>Level 1A</p> <p>Algorithms and Programming 1A-AP-08 Model daily processes by creating and following algorithms (sets of step-by-step instructions) to complete tasks.</p> <p>1A-AP-09 Model the way programs store and manipulate data by using numbers or other symbols to represent information.</p> <p>1A-AP-10 Develop programs with sequences and simple loops, to express ideas or address a problem.</p>

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
		<p>Information in the real world can be represented in computer programs. Programs store and manipulate data, such as numbers, words, colors, and images. The type of data determines the actions and attributes associated with it.</p> <p><u>K-2. Algorithms and Programming. Control</u> Computers follow precise sequences of instructions that automate tasks. Program execution can also be nonsequential by repeating patterns of instructions and using events to initiate instructions.</p> <p><u>K-2. Algorithms and Programming. Program Development</u> People develop programs collaboratively and for a purpose, such as expressing ideas or addressing problems.</p>	<p>1A-AP-11 Decompose (break down) the steps needed to solve a problem into a precise sequence of instructions.</p> <p>1A-AP-12 Develop plans that describe a program's sequence of events, goals, and expected outcomes.</p>
Simple inputs: Catch the fish	In this lesson, pupils will consolidate understanding of start events and	click events and combine these to create a fishing game.	
Links	<p><u>Programming and development</u> Knows that users can develop their own programs, and can demonstrate this by creating a simple program in an environment that does not rely on text. (AL) Executes, checks and changes programs. (AL) Understands that programs execute by following precise instructions. (AL)</p>	<p><u>K-2. Computing Systems. Devices</u> People use computing devices to perform a variety of tasks accurately and quickly. Computing devices interpret and follow the instructions they are given literally.</p> <p><u>K-2. Computing Systems. Troubleshooting</u> Computing systems might not work as expected because of hardware or software problems. Clearly describing a problem is the first step toward finding a solution.</p> <p><u>K-2. Algorithms and Programming. Algorithms</u> People follow and create processes as part of daily life. Many of these processes can be expressed as algorithms that computers can follow.</p> <p><u>K-2. Algorithms and Programming. Variables</u> Information in the real world can be represented in computer programs. Programs store and manipulate data, such as numbers, words, colors, and images. The type of data determines the actions and attributes associated with it.</p> <p><u>K-2. Algorithms and Programming. Control</u> Computers follow precise sequences of instructions that automate tasks. Program execution can also be nonsequential by repeating patterns of instructions and using events to initiate instructions.</p> <p><u>K-2. Algorithms and Programming. Program Development</u> People develop programs collaboratively and for a purpose, such as expressing ideas or addressing problems.</p>	<p><u>Level 1A</u></p> <p><u>Algorithms and Programming</u> 1A-AP-08 Model daily processes by creating and following algorithms (sets of step-by-step instructions) to complete tasks.</p> <p>1A-AP-09 Model the way programs store and manipulate data by using numbers or other symbols to represent information.</p> <p>1A-AP-10 Develop programs with sequences and simple loops, to express ideas or address a problem.</p> <p>1A-AP-11 Decompose (break down) the steps needed to solve a problem into a precise sequence of instructions.</p> <p>1A-AP-12 Develop plans that describe a program's sequence of events, goals, and expected outcomes.</p>
Simple inputs: Magic castle	To combine start events and click events in code to make a magic castle scene.		
Links	<u>Programming and development</u>	<u>K-2. Computing Systems. Devices</u>	<u>Level 1A</u>

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
	<p>Knows that users can develop their own programs, and can demonstrate this by creating a simple program in an environment that does not rely on text. (AL)</p> <p>Executes, checks and changes programs. (AL)</p> <p>Understands that programs execute by following precise instructions. (AL)</p>	<p>People use computing devices to perform a variety of tasks accurately and quickly. Computing devices interpret and follow the instructions they are given literally.</p> <p><u>K-2. Computing Systems. Troubleshooting</u></p> <p>Computing systems might not work as expected because of hardware or software problems. Clearly describing a problem is the first step toward finding a solution.</p> <p><u>K-2. Algorithms and Programming. Algorithms</u></p> <p>People follow and create processes as part of daily life. Many of these processes can be expressed as algorithms that computers can follow.</p> <p><u>K-2. Algorithms and Programming. Variables</u></p> <p>Information in the real world can be represented in computer programs. Programs store and manipulate data, such as numbers, words, colors, and images. The type of data determines the actions and attributes associated with it.</p> <p><u>K-2. Algorithms and Programming. Control</u></p> <p>Computers follow precise sequences of instructions that automate tasks. Program execution can also be nonsequential by repeating patterns of instructions and using events to initiate instructions.</p> <p><u>K-2. Algorithms and Programming. Program Development</u></p> <p>People develop programs collaboratively and for a purpose, such as expressing ideas or addressing problems.</p>	<p><u>Algorithms and Programming</u></p> <p>1A-AP-08 Model daily processes by creating and following algorithms (sets of step-by-step instructions) to complete tasks.</p> <p>1A-AP-09 Model the way programs store and manipulate data by using numbers or other symbols to represent information.</p> <p>1A-AP-10 Develop programs with sequences and simple loops, to express ideas or address a problem.</p> <p>1A-AP-11 Decompose (break down) the steps needed to solve a problem into a precise sequence of instructions.</p> <p>1A-AP-12 Develop plans that describe a program’s sequence of events, goals, and expected outcomes.</p>
<p>Simple inputs: Emergency</p>	<p>To combine start events and click events to program cars and emergency vehicles in an animated traffic scene.</p>		
<p>Links</p>	<p><u>Programming and development</u></p> <p>Knows that users can develop their own programs, and can demonstrate this by creating a simple program in an environment that does not rely on text. (AL)</p> <p>Executes, checks and changes programs. (AL)</p> <p>Understands that programs execute by following precise instructions. (AL)</p>	<p><u>K-2. Computing Systems. Devices</u></p> <p>People use computing devices to perform a variety of tasks accurately and quickly. Computing devices interpret and follow the instructions they are given literally.</p> <p><u>K-2. Computing Systems. Troubleshooting</u></p> <p>Computing systems might not work as expected because of hardware or software problems. Clearly describing a problem is the first step toward finding a solution.</p> <p><u>K-2. Algorithms and Programming. Algorithms</u></p> <p>People follow and create processes as part of daily life. Many of these processes can be expressed as algorithms that computers can follow.</p> <p><u>K-2. Algorithms and Programming. Variables</u></p> <p>Information in the real world can be represented in computer programs. Programs store and manipulate data, such as numbers, words, colors, and images. The type of data determines the actions and attributes associated with it.</p>	<p><u>Level 1A</u></p> <p><u>Algorithms and Programming</u></p> <p>1A-AP-08 Model daily processes by creating and following algorithms (sets of step-by-step instructions) to complete tasks.</p> <p>1A-AP-09 Model the way programs store and manipulate data by using numbers or other symbols to represent information.</p> <p>1A-AP-10 Develop programs with sequences and simple loops, to express ideas or address a problem.</p> <p>1A-AP-11 Decompose (break down) the steps needed to solve a problem into a precise sequence of instructions.</p>

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
		<p><u>K-2. Algorithms and Programming. Control</u> Computers follow precise sequences of instructions that automate tasks. Program execution can also be nonsequential by repeating patterns of instructions and using events to initiate instructions.</p> <p><u>K-2. Algorithms and Programming. Program Development</u> People develop programs collaboratively and for a purpose, such as expressing ideas or addressing problems.</p>	<p>1A-AP-12 Develop plans that describe a program’s sequence of events, goals, and expected outcomes.</p>
Block coding: Level 2			
Different sorts of input: Red Riding Hood	To write code that makes an object move around the screen when keys are pressed.		
Links	<p><u>Programming and development</u> Knows that users can develop their own programs, and can demonstrate this by creating a simple program in an environment that does not rely on text. (AL) Executes, checks and changes programs. (AL) Understands that programs execute by following precise instructions. (AL) <u>Algorithms</u> Understands what an algorithm is and is able to express simple linear (non-branching) algorithms symbolically. (AL)</p>	<p><u>K-2. Computing Systems. Devices</u> People use computing devices to perform a variety of tasks accurately and quickly. Computing devices interpret and follow the instructions they are given literally.</p> <p><u>K-2. Computing Systems. Troubleshooting</u> Computing systems might not work as expected because of hardware or software problems. Clearly describing a problem is the first step toward finding a solution.</p> <p><u>K-2. Algorithms and Programming. Algorithms</u> People follow and create processes as part of daily life. Many of these processes can be expressed as algorithms that computers can follow.</p> <p><u>K-2. Algorithms and Programming. Variables</u> Information in the real world can be represented in computer programs. Programs store and manipulate data, such as numbers, words, colors, and images. The type of data determines the actions and attributes associated with it.</p> <p><u>K-2. Algorithms and Programming. Control</u> Computers follow precise sequences of instructions that automate tasks. Program execution can also be nonsequential by repeating patterns of instructions and using events to initiate instructions.</p> <p><u>K-2. Algorithms and Programming. Modularity</u> Complex tasks can be broken down into simpler instructions, some of which can be broken down even further. Likewise, instructions can be combined to accomplish complex tasks.</p> <p><u>K-2. Algorithms and Programming. Program Development</u> People develop programs collaboratively and for a purpose, such as expressing ideas or addressing problems.</p>	<p><u>Level 1A</u></p> <p><u>Algorithms and Programming</u> 1A-AP-08 Model daily processes by creating and following algorithms (sets of step-by-step instructions) to complete tasks.</p> <p>1A-AP-09 Model the way programs store and manipulate data by using numbers or other symbols to represent information.</p> <p>1A-AP-10 Develop programs with sequences and simple loops, to express ideas or address a problem.</p> <p>1A-AP-11 Decompose (break down) the steps needed to solve a problem into a precise sequence of instructions.</p> <p>1A-AP-12 Develop plans that describe a program’s sequence of events, goals, and expected outcomes.</p>
Different sorts of input: Key to the race	To make objects perform different actions when keys are pressed on the keyboard.		

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
Links	<p><u>Programming and development</u> Knows that users can develop their own programs, and can demonstrate this by creating a simple program in an environment that does not rely on text. (AL) Executes, checks and changes programs. (AL) Understands that programs execute by following precise instructions. (AL)</p>	<p><u>K-2. Computing Systems. Devices</u> People use computing devices to perform a variety of tasks accurately and quickly. Computing devices interpret and follow the instructions they are given literally. <u>K-2. Computing Systems. Troubleshooting</u> Computing systems might not work as expected because of hardware or software problems. Clearly describing a problem is the first step toward finding a solution. <u>K-2. Algorithms and Programming. Algorithms</u> People follow and create processes as part of daily life. Many of these processes can be expressed as algorithms that computers can follow. <u>K-2. Algorithms and Programming. Variables</u> Information in the real world can be represented in computer programs. Programs store and manipulate data, such as numbers, words, colors, and images. The type of data determines the actions and attributes associated with it. <u>K-2. Algorithms and Programming. Control</u> Computers follow precise sequences of instructions that automate tasks. Program execution can also be nonsequential by repeating patterns of instructions and using events to initiate instructions. <u>K-2. Algorithms and Programming. Modularity</u> Complex tasks can be broken down into simpler instructions, some of which can be broken down even further. Likewise, instructions can be combined to accomplish complex tasks. <u>K-2. Algorithms and Programming. Program Development</u> People develop programs collaboratively and for a purpose, such as expressing ideas or addressing problems.</p>	<p><u>Level 1A</u> <u>Algorithms and Programming</u> 1A-AP-08 Model daily processes by creating and following algorithms (sets of step-by-step instructions) to complete tasks. 1A-AP-09 Model the way programs store and manipulate data by using numbers or other symbols to represent information. 1A-AP-10 Develop programs with sequences and simple loops, to express ideas or address a problem. 1A-AP-11 Decompose (break down) the steps needed to solve a problem into a precise sequence of instructions. 1A-AP-12 Develop plans that describe a program’s sequence of events, goals, and expected outcomes.</p>
Different sorts of input: Up in the air	To write code that makes an object change direction when different keys on the keyboard are pressed.		
Links	<p><u>Programming and development</u> Knows that users can develop their own programs, and can demonstrate this by creating a simple program in an environment that does not rely on text. (AL) Executes, checks and changes programs. (AL) Understands that programs execute by following precise instructions. (AL) <u>Algorithms</u> Understands what an algorithm is and is able to express simple linear (non-branching) algorithms symbolically. (AL)</p>	<p><u>K-2. Algorithms and Programming. Algorithms</u> People follow and create processes as part of daily life. Many of these processes can be expressed as algorithms that computers can follow. <u>K-2. Algorithms and Programming. Variables</u> Information in the real world can be represented in computer programs. Programs store and manipulate data, such as numbers, words, colors, and images. The type of data determines the actions and attributes associated with it. <u>K-2. Algorithms and Programming. Control</u></p>	<p><u>Level 1A</u> <u>Algorithms and Programming</u> 1A-AP-08 Model daily processes by creating and following algorithms (sets of step-by-step instructions) to complete tasks. 1A-AP-09 Model the way programs store and manipulate data by using numbers or other symbols to represent information.</p>

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
		<p>Computers follow precise sequences of instructions that automate tasks. Program execution can also be nonsequential by repeating patterns of instructions and using events to initiate instructions.</p> <p><u>K-2. Algorithms and Programming. Modularity</u> Complex tasks can be broken down into simpler instructions, some of which can be broken down even further. Likewise, instructions can be combined to accomplish complex tasks.</p> <p><u>K-2. Algorithms and Programming. Program Development</u> People develop programs collaboratively and for a purpose, such as expressing ideas or addressing problems.</p>	<p>1A-AP-10 Develop programs with sequences and simple loops, to express ideas or address a problem.</p> <p>1A-AP-11 Decompose (break down) the steps needed to solve a problem into a precise sequence of instructions.</p> <p>1A-AP-12 Develop plans that describe a program’s sequence of events, goals, and expected outcomes.</p>
<p>Different sorts of input: Shark attack</p>	<p>To write code that makes an object change direction when the pointer is pressed and released.</p>		
<p>Links</p>	<p><u>Programming and development</u> Knows that users can develop their own programs, and can demonstrate this by creating a simple program in an environment that does not rely on text. (AL) Executes, checks and changes programs. (AL) Understands that programs execute by following precise instructions. (AL)</p>	<p><u>K-2. Algorithms and Programming. Algorithms</u> People follow and create processes as part of daily life. Many of these processes can be expressed as algorithms that computers can follow.</p> <p><u>K-2. Algorithms and Programming. Variables</u> Information in the real world can be represented in computer programs. Programs store and manipulate data, such as numbers, words, colors, and images. The type of data determines the actions and attributes associated with it.</p> <p><u>K-2. Algorithms and Programming. Control</u> Computers follow precise sequences of instructions that automate tasks. Program execution can also be nonsequential by repeating patterns of instructions and using events to initiate instructions.</p> <p><u>K-2. Algorithms and Programming. Modularity</u> Complex tasks can be broken down into simpler instructions, some of which can be broken down even further. Likewise, instructions can be combined to accomplish complex tasks.</p> <p><u>K-2. Algorithms and Programming. Program Development</u> People develop programs collaboratively and for a purpose, such as expressing ideas or addressing problems.</p>	<p><u>Level 1A</u></p> <p><u>Algorithms and Programming</u> 1A-AP-08 Model daily processes by creating and following algorithms (sets of step-by-step instructions) to complete tasks.</p> <p>1A-AP-09 Model the way programs store and manipulate data by using numbers or other symbols to represent information.</p> <p>1A-AP-10 Develop programs with sequences and simple loops, to express ideas or address a problem.</p> <p>1A-AP-11 Decompose (break down) the steps needed to solve a problem into a precise sequence of instructions.</p> <p>1A-AP-12 Develop plans that describe a program’s sequence of events, goals, and expected outcomes.</p>
<p>Different sorts of input: Snow White</p>	<p>To write code where different inputs can be used to make objects move and disappear.</p>		
<p>Links</p>	<p><u>Programming and development</u> Knows that users can develop their own programs, and can demonstrate this by creating a simple program in an environment that does not rely on text. (AL) Executes, checks and changes programs. (AL)</p>	<p><u>K-2. Algorithms and Programming. Algorithms</u> People follow and create processes as part of daily life. Many of these processes can be expressed as algorithms that computers can follow.</p> <p><u>K-2. Algorithms and Programming. Variables</u> Information in the real world can be represented in computer programs. Programs store and manipulate data, such as numbers, words, colors,</p>	<p><u>Level 1A</u></p> <p><u>Algorithms and Programming</u> 1A-AP-08 Model daily processes by creating and following algorithms (sets of step-by-step instructions) to complete tasks.</p>

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
	Understands that programs execute by following precise instructions. (AL)	and images. The type of data determines the actions and attributes associated with it. <u>K-2. Algorithms and Programming. Control</u> Computers follow precise sequences of instructions that automate tasks. Program execution can also be nonsequential by repeating patterns of instructions and using events to initiate instructions. <u>K-2. Algorithms and Programming. Modularity</u> Complex tasks can be broken down into simpler instructions, some of which can be broken down even further. Likewise, instructions can be combined to accomplish complex tasks. <u>K-2. Algorithms and Programming. Program Development</u> People develop programs collaboratively and for a purpose, such as expressing ideas or addressing problems.	1A-AP-09 Model the way programs store and manipulate data by using numbers or other symbols to represent information. 1A-AP-10 Develop programs with sequences and simple loops, to express ideas or address a problem. 1A-AP-11 Decompose (break down) the steps needed to solve a problem into a precise sequence of instructions. 1A-AP-12 Develop plans that describe a program’s sequence of events, goals, and expected outcomes.
Buttons and instructions: Fly a helicopter	To write code where buttons can be used to make an object move around the screen.		
Links	<u>Programming and development</u> Knows that users can develop their own programs, and can demonstrate this by creating a simple program in an environment that does not rely on text. (AL) Executes, checks and changes programs. (AL) Understands that programs execute by following precise instructions. (AL) <u>Algorithms</u> Understands what an algorithm is and is able to express simple linear (non-branching) algorithms symbolically. (AL)	<u>K-2. Algorithms and Programming. Algorithms</u> People follow and create processes as part of daily life. Many of these processes can be expressed as algorithms that computers can follow. <u>K-2. Algorithms and Programming. Variables</u> Information in the real world can be represented in computer programs. Programs store and manipulate data, such as numbers, words, colors, and images. The type of data determines the actions and attributes associated with it. <u>K-2. Algorithms and Programming. Control</u> Computers follow precise sequences of instructions that automate tasks. Program execution can also be nonsequential by repeating patterns of instructions and using events to initiate instructions. <u>K-2. Algorithms and Programming. Modularity</u> Complex tasks can be broken down into simpler instructions, some of which can be broken down even further. Likewise, instructions can be combined to accomplish complex tasks. <u>K-2. Algorithms and Programming. Program Development</u> People develop programs collaboratively and for a purpose, such as expressing ideas or addressing problems.	<u>Level 1A</u> <u>Algorithms and Programming</u> 1A-AP-08 Model daily processes by creating and following algorithms (sets of step-by-step instructions) to complete tasks. 1A-AP-09 Model the way programs store and manipulate data by using numbers or other symbols to represent information. 1A-AP-10 Develop programs with sequences and simple loops, to express ideas or address a problem. 1A-AP-11 Decompose (break down) the steps needed to solve a problem into a precise sequence of instructions. 1A-AP-12 Develop plans that describe a program’s sequence of events, goals, and expected outcomes.
Buttons and instructions: Slug hunt	To write the code for a simple game where buttons are used to move an object around.		
Links	<u>Programming and development</u>	<u>K-2. Algorithms and Programming. Algorithms</u>	<u>Level 1A</u>

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
	<p>Knows that users can develop their own programs, and can demonstrate this by creating a simple program in an environment that does not rely on text. (AL)</p> <p>Executes, checks and changes programs. (AL)</p> <p>Understands that programs execute by following precise instructions. (AL)</p> <p><u>Algorithms</u></p> <p>Understands what an algorithm is and is able to express simple linear (non-branching) algorithms symbolically. (AL)</p> <p>Detects and corrects errors, i.e. debugging in algorithms. (AL)</p>	<p>People follow and create processes as part of daily life. Many of these processes can be expressed as algorithms that computers can follow.</p> <p><u>K-2. Algorithms and Programming. Variables</u></p> <p>Information in the real world can be represented in computer programs. Programs store and manipulate data, such as numbers, words, colors, and images. The type of data determines the actions and attributes associated with it.</p> <p><u>K-2. Algorithms and Programming. Control</u></p> <p>Computers follow precise sequences of instructions that automate tasks. Program execution can also be nonsequential by repeating patterns of instructions and using events to initiate instructions.</p> <p><u>K-2. Algorithms and Programming. Modularity</u></p> <p>Complex tasks can be broken down into simpler instructions, some of which can be broken down even further. Likewise, instructions can be combined to accomplish complex tasks.</p> <p><u>K-2. Algorithms and Programming. Program Development</u></p> <p>People develop programs collaboratively and for a purpose, such as expressing ideas or addressing problems.</p>	<p><u>Algorithms and Programming</u></p> <p>1A-AP-08 Model daily processes by creating and following algorithms (sets of step-by-step instructions) to complete tasks.</p> <p>1A-AP-09 Model the way programs store and manipulate data by using numbers or other symbols to represent information.</p> <p>1A-AP-10 Develop programs with sequences and simple loops, to express ideas or address a problem.</p> <p>1A-AP-11 Decompose (break down) the steps needed to solve a problem into a precise sequence of instructions.</p> <p>1A-AP-12 Develop plans that describe a program’s sequence of events, goals, and expected outcomes.</p>
<p>Buttons and instructions: Find my cat!</p>	<p>To write the code for a simple game where buttons are used to move an object around and cast a magic ‘disappearing spell’.</p>		
<p>Links</p>	<p><u>Programming and development</u></p> <p>Knows that users can develop their own programs, and can demonstrate this by creating a simple program in an environment that does not rely on text. (AL)</p> <p>Executes, checks and changes programs. (AL)</p> <p>Understands that programs execute by following precise instructions. (AL)</p> <p>Demonstrates care and precision to avoid errors. (AL)</p> <p><u>Algorithms</u></p> <p>Understands what an algorithm is and is able to express simple linear (non-branching) algorithms symbolically. (AL)</p> <p>Detects and corrects errors, i.e. debugging in algorithms. (AL)</p>	<p><u>K-2. Algorithms and Programming. Algorithms</u></p> <p>People follow and create processes as part of daily life. Many of these processes can be expressed as algorithms that computers can follow.</p> <p><u>K-2. Algorithms and Programming. Variables</u></p> <p>Information in the real world can be represented in computer programs. Programs store and manipulate data, such as numbers, words, colors, and images. The type of data determines the actions and attributes associated with it.</p> <p><u>K-2. Algorithms and Programming. Control</u></p> <p>Computers follow precise sequences of instructions that automate tasks. Program execution can also be nonsequential by repeating patterns of instructions and using events to initiate instructions.</p> <p><u>K-2. Algorithms and Programming. Modularity</u></p> <p>Complex tasks can be broken down into simpler instructions, some of which can be broken down even further. Likewise, instructions can be combined to accomplish complex tasks.</p> <p><u>K-2. Algorithms and Programming. Program Development</u></p> <p>People develop programs collaboratively and for a purpose, such as expressing ideas or addressing problems.</p>	<p><u>Level 1A</u></p> <p><u>Algorithms and Programming</u></p> <p>1A-AP-08 Model daily processes by creating and following algorithms (sets of step-by-step instructions) to complete tasks.</p> <p>1A-AP-09 Model the way programs store and manipulate data by using numbers or other symbols to represent information.</p> <p>1A-AP-10 Develop programs with sequences and simple loops, to express ideas or address a problem.</p> <p>1A-AP-11 Decompose (break down) the steps needed to solve a problem into a precise sequence of instructions.</p> <p>1A-AP-12 Develop plans that describe a program’s sequence of events, goals, and expected outcomes.</p>

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
Buttons and instructions: Hungry Migbod	To write code where buttons are used to move a monster around and eat (hide) fruit.		
Links	<p><u>Programming and development</u> Knows that users can develop their own programs, and can demonstrate this by creating a simple program in an environment that does not rely on text. (AL) Executes, checks and changes programs. (AL) Understands that programs execute by following precise instructions. (AL) <u>Algorithms</u> Understands what an algorithm is and is able to express simple linear (non-branching) algorithms symbolically. (AL) Detects and corrects errors, i.e. debugging in algorithms. (AL)</p>	<p><u>K–2. Algorithms and Programming. Algorithms</u> People follow and create processes as part of daily life. Many of these processes can be expressed as algorithms that computers can follow. <u>K–2. Algorithms and Programming. Variables</u> Information in the real world can be represented in computer programs. Programs store and manipulate data, such as numbers, words, colors, and images. The type of data determines the actions and attributes associated with it. <u>K–2. Algorithms and Programming. Control</u> Computers follow precise sequences of instructions that automate tasks. Program execution can also be nonsequential by repeating patterns of instructions and using events to initiate instructions. <u>K–2. Algorithms and Programming. Modularity</u> Complex tasks can be broken down into simpler instructions, some of which can be broken down even further. Likewise, instructions can be combined to accomplish complex tasks. <u>K–2. Algorithms and Programming. Program Development</u> People develop programs collaboratively and for a purpose, such as expressing ideas or addressing problems.</p>	<p><u>Level 1A</u> <u>Algorithms and Programming</u> 1A-AP-08 Model daily processes by creating and following algorithms (sets of step-by-step instructions) to complete tasks. 1A-AP-09 Model the way programs store and manipulate data by using numbers or other symbols to represent information. 1A-AP-10 Develop programs with sequences and simple loops, to express ideas or address a problem. 1A-AP-11 Decompose (break down) the steps needed to solve a problem into a precise sequence of instructions. 1A-AP-12 Develop plans that describe a program’s sequence of events, goals, and expected outcomes.</p>
Block coding: Level 3			
Sequence and animation: Stepping through space	To write a computer program where different pieces of code execute in a particular sequence.		
Links	<p><u>Programming and development</u> Knows that users can develop their own programs, and can demonstrate this by creating a simple program in an environment that does not rely on text. (AL) Executes, checks and changes programs. (AL) Understands that programs execute by following precise instructions. (AL) <u>Algorithms</u> Understands what an algorithm is and is able to express simple linear (non-branching) algorithms symbolically. (AL) Detects and corrects errors, i.e. debugging in algorithms. (AL)</p>	<p><u>3–5. Algorithms and Programming. Algorithms</u> Different algorithms can achieve the same result. Some algorithms are more appropriate for a specific context than others. <u>3–5. Algorithms and Programming. Control</u> Control structures, including loops, event handlers, and conditionals, are used to specify the flow of execution. Conditionals selectively execute or skip instructions under different conditions. <u>3–5. Algorithms and Programming. Modularity</u> Programs can be broken down into smaller parts to facilitate their design, implementation, and review. Programs can also be created by incorporating smaller portions of programs that have already been created. <u>3–5. Algorithms and Programming. Program Development</u></p>	<p><u>Level 1B</u> <u>Algorithms and Programming</u> 1B-AP-08 Compare and refine multiple algorithms for the same task and determine which is the most appropriate. 1B-AP-10 Create programs that include sequences, events, loops, and conditionals. 1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.</p>

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
		<p>People develop programs using an iterative process involving design, implementation, and review. Design often involves reusing existing code or remixing other programs within a community. People continuously review whether programs work as expected, and they fix, or debug, parts that do not. Repeating these steps enables people to refine and improve programs.</p>	<p>1B-AP-12 Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.</p> <p>1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences</p> <p>1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs.</p> <p>1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.</p> <p>1B-AP-16 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development.</p> <p>1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.</p>
<p>Sequence and animation: Snail vs spider</p>	<p>To create a program that uses sequences for two different objects moving on the screen.</p>		
<p>Links</p>	<p><u>Programming and development</u> Knows that users can develop their own programs, and can demonstrate this by creating a simple program in an environment that does not rely on text. (AL) Executes, checks and changes programs. (AL) Understands that programs execute by following precise instructions. (AL) <u>Algorithms</u> Understands what an algorithm is and is able to express simple linear (non-branching) algorithms symbolically. (AL) Detects and corrects errors, i.e. debugging in algorithms. (AL)</p>	<p><u>3-5. Algorithms and Programming. Algorithms</u> Different algorithms can achieve the same result. Some algorithms are more appropriate for a specific context than others. <u>3-5. Algorithms and Programming. Control</u> Control structures, including loops, event handlers, and conditionals, are used to specify the flow of execution. Conditionals selectively execute or skip instructions under different conditions. <u>3-5. Algorithms and Programming. Modularity</u> Programs can be broken down into smaller parts to facilitate their design, implementation, and review. Programs can also be created by incorporating smaller portions of programs that have already been created. <u>3-5. Algorithms and Programming. Program Development</u> People develop programs using an iterative process involving design, implementation, and review. Design often involves reusing existing code or remixing other programs within a community. People continuously</p>	<p><u>Level 1B</u> <u>Algorithms and Programming</u> 1B-AP-08 Compare and refine multiple algorithms for the same task and determine which is the most appropriate. 1B-AP-10 Create programs that include sequences, events, loops, and conditionals. 1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process. 1B-AP-12 Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.</p>

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
		review whether programs work as expected, and they fix, or debug, parts that do not. Repeating these steps enables people to refine and improve programs.	<p>1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences</p> <p>1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs.</p> <p>1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.</p> <p>1B-AP-16 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development.</p> <p>1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.</p>
Sequence and animation: Alien space race	To write code that uses a timer to create a sequence of events.		
Links	<p><u>Programming and development</u> Knows that users can develop their own programs, and can demonstrate this by creating a simple program in an environment that does not rely on text. (AL) Executes, checks and changes programs. (AL) Understands that programs execute by following precise instructions. (AL)</p> <p><u>Algorithms</u> Understands what an algorithm is and is able to express simple linear (non-branching) algorithms symbolically. (AL) Detects and corrects errors, i.e. debugging in algorithms. (AL)</p>	<p><u>3–5. Algorithms and Programming. Algorithms</u> Different algorithms can achieve the same result. Some algorithms are more appropriate for a specific context than others.</p> <p><u>3–5. Algorithms and Programming. Control</u> Control structures, including loops, event handlers, and conditionals, are used to specify the flow of execution. Conditionals selectively execute or skip instructions under different conditions.</p> <p><u>3–5. Algorithms and Programming. Modularity</u> Programs can be broken down into smaller parts to facilitate their design, implementation, and review. Programs can also be created by incorporating smaller portions of programs that have already been created.</p> <p><u>3–5. Algorithms and Programming. Program Development</u> People develop programs using an iterative process involving design, implementation, and review. Design often involves reusing existing code or remixing other programs within a community. People continuously review whether programs work as expected, and they fix, or debug, parts that do not. Repeating these steps enables people to refine and improve programs.</p>	<p><u>Level 1B</u></p> <p><u>Algorithms and Programming</u></p> <p>1B-AP-08 Compare and refine multiple algorithms for the same task and determine which is the most appropriate.</p> <p>1B-AP-10 Create programs that include sequences, events, loops, and conditionals.</p> <p>1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.</p> <p>1B-AP-12 Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.</p>

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
			<p>1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences</p> <p>1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs.</p> <p>1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.</p> <p>1B-AP-16 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development.</p> <p>1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.</p>
Sequence and animation: Traffic lights	To write code that uses a timer to create a sequence of traffic lights turning on and off.		
Links	<p><u>Programming and development</u> Knows that users can develop their own programs, and can demonstrate this by creating a simple program in an environment that does not rely on text. (AL) Executes, checks and changes programs. (AL) Understands that programs execute by following precise instructions. (AL) <u>Algorithms</u> Understands what an algorithm is and is able to express simple linear (non-branching) algorithms symbolically. (AL) Detects and corrects errors, i.e. debugging in algorithms. (AL)</p>	<p><u>3–5. Algorithms and Programming. Algorithms</u> Different algorithms can achieve the same result. Some algorithms are more appropriate for a specific context than others. <u>3–5. Algorithms and Programming. Control</u> Control structures, including loops, event handlers, and conditionals, are used to specify the flow of execution. Conditionals selectively execute or skip instructions under different conditions. <u>3–5. Algorithms and Programming. Modularity</u> Programs can be broken down into smaller parts to facilitate their design, implementation, and review. Programs can also be created by incorporating smaller portions of programs that have already been created. <u>3–5. Algorithms and Programming. Program Development</u> People develop programs using an iterative process involving design, implementation, and review. Design often involves reusing existing code or remixing other programs within a community. People continuously review whether programs work as expected, and they fix, or debug, parts that do not. Repeating these steps enables people to refine and improve programs.</p>	<p><u>Level 1B</u></p> <p><u>Algorithms and Programming</u></p> <p>1B-AP-08 Compare and refine multiple algorithms for the same task and determine which is the most appropriate.</p> <p>1B-AP-10 Create programs that include sequences, events, loops, and conditionals.</p> <p>1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.</p> <p>1B-AP-12 Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.</p> <p>1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences</p>

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
			<p>1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs.</p> <p>1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.</p> <p>1B-AP-16 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development.</p> <p>1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.</p>
Conditional events: Space maze	To use 'hit events' to program a space maze game in which an object reacts to particular conditions.		
Links	<p><u>Programming and development</u> Knows that users can develop their own programs, and can demonstrate this by creating a simple program in an environment that does not rely on text. (AL) Executes, checks and changes programs. (AL) Understands that programs execute by following precise instructions. (AL) <u>Algorithms</u> Understands what an algorithm is and is able to express simple linear (non-branching) algorithms symbolically. (AL) Detects and corrects errors, i.e. debugging in algorithms. (AL)</p>	<p><u>3-5. Algorithms and Programming. Algorithms</u> Different algorithms can achieve the same result. Some algorithms are more appropriate for a specific context than others.</p> <p><u>3-5. Algorithms and Programming. Control</u> Control structures, including loops, event handlers, and conditionals, are used to specify the flow of execution. Conditionals selectively execute or skip instructions under different conditions.</p> <p><u>3-5. Algorithms and Programming. Modularity</u> Programs can be broken down into smaller parts to facilitate their design, implementation, and review. Programs can also be created by incorporating smaller portions of programs that have already been created.</p> <p><u>3-5. Algorithms and Programming. Program Development</u> People develop programs using an iterative process involving design, implementation, and review. Design often involves reusing existing code or remixing other programs within a community. People continuously review whether programs work as expected, and they fix, or debug, parts that do not. Repeating these steps enables people to refine and improve programs.</p>	<p><u>Level 1B</u></p> <p><u>Algorithms and Programming</u></p> <p>1B-AP-08 Compare and refine multiple algorithms for the same task and determine which is the most appropriate.</p> <p>1B-AP-10 Create programs that include sequences, events, loops, and conditionals.</p> <p>1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.</p> <p>1B-AP-12 Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.</p> <p>1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences</p> <p>1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs.</p>

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
			<p>1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.</p> <p>1B-AP-16 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development.</p> <p>1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.</p>
<p>Conditional events: Self-driving car</p>	<p>To use conditional hit events to control the movement of a car on the screen.</p>		
<p>Links</p>	<p><u>Programming and development</u> Knows that users can develop their own programs, and can demonstrate this by creating a simple program in an environment that does not rely on text. (AL) Executes, checks and changes programs. (AL) Understands that programs execute by following precise instructions. (AL) <u>Algorithms</u> Understands what an algorithm is and is able to express simple linear (non-branching) algorithms symbolically. (AL) Detects and corrects errors, i.e. debugging in algorithms. (AL)</p>	<p><u>3–5. Algorithms and Programming. Algorithms</u> Different algorithms can achieve the same result. Some algorithms are more appropriate for a specific context than others. <u>3–5. Algorithms and Programming. Control</u> Control structures, including loops, event handlers, and conditionals, are used to specify the flow of execution. Conditionals selectively execute or skip instructions under different conditions. <u>3–5. Algorithms and Programming. Modularity</u> Programs can be broken down into smaller parts to facilitate their design, implementation, and review. Programs can also be created by incorporating smaller portions of programs that have already been created. <u>3–5. Algorithms and Programming. Program Development</u> People develop programs using an iterative process involving design, implementation, and review. Design often involves reusing existing code or remixing other programs within a community. People continuously review whether programs work as expected, and they fix, or debug, parts that do not. Repeating these steps enables people to refine and improve programs.</p>	<p><u>Level 1B</u> <u>Algorithms and Programming</u></p> <p>1B-AP-08 Compare and refine multiple algorithms for the same task and determine which is the most appropriate.</p> <p>1B-AP-10 Create programs that include sequences, events, loops, and conditionals.</p> <p>1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.</p> <p>1B-AP-12 Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.</p> <p>1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences</p> <p>1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs.</p> <p>1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.</p>

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
			<p>1B-AP-16 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development.</p> <p>1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.</p>
<p>Conditional events: Hungry snake</p>	<p>To make a simple game that uses conditional hit events to check if one object has hit another.</p>		
<p>Links</p>	<p><u>Programming and development</u> Knows that users can develop their own programs, and can demonstrate this by creating a simple program in an environment that does not rely on text. (AL) Executes, checks and changes programs. (AL) Understands that programs execute by following precise instructions. (AL) <u>Algorithms</u> Understands what an algorithm is and is able to express simple linear (non-branching) algorithms symbolically. (AL) Detects and corrects errors, i.e. debugging in algorithms. (AL)</p>	<p><u>3–5. Algorithms and Programming. Algorithms</u> Different algorithms can achieve the same result. Some algorithms are more appropriate for a specific context than others. <u>3–5. Algorithms and Programming. Control</u> Control structures, including loops, event handlers, and conditionals, are used to specify the flow of execution. Conditionals selectively execute or skip instructions under different conditions. <u>3–5. Algorithms and Programming. Modularity</u> Programs can be broken down into smaller parts to facilitate their design, implementation, and review. Programs can also be created by incorporating smaller portions of programs that have already been created. <u>3–5. Algorithms and Programming. Program Development</u> People develop programs using an iterative process involving design, implementation, and review. Design often involves reusing existing code or remixing other programs within a community. People continuously review whether programs work as expected, and they fix, or debug, parts that do not. Repeating these steps enables people to refine and improve programs.</p>	<p><u>Level 1B</u></p> <p><u>Algorithms and Programming</u></p> <p>1B-AP-08 Compare and refine multiple algorithms for the same task and determine which is the most appropriate.</p> <p>1B-AP-10 Create programs that include sequences, events, loops, and conditionals.</p> <p>1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.</p> <p>1B-AP-12 Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.</p> <p>1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences</p> <p>1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs.</p> <p>1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.</p> <p>1B-AP-16 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development.</p>

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
			<p>1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.</p>
<p>Conditional events: Pufferfish pop</p>	<p>To program a simple game where conditional events are used to check whether objects have collided.</p>		
<p>Links</p>	<p><u>Programming and development</u> Knows that users can develop their own programs, and can demonstrate this by creating a simple program in an environment that does not rely on text. (AL) Executes, checks and changes programs. (AL) Understands that programs execute by following precise instructions. (AL) <u>Algorithms</u> Understands what an algorithm is and is able to express simple linear (non-branching) algorithms symbolically. (AL) Detects and corrects errors, i.e. debugging in algorithms. (AL)</p>	<p><u>3–5. Algorithms and Programming. Algorithms</u> Different algorithms can achieve the same result. Some algorithms are more appropriate for a specific context than others. <u>3–5. Algorithms and Programming. Control</u> Control structures, including loops, event handlers, and conditionals, are used to specify the flow of execution. Conditionals selectively execute or skip instructions under different conditions. <u>3–5. Algorithms and Programming. Modularity</u> Programs can be broken down into smaller parts to facilitate their design, implementation, and review. Programs can also be created by incorporating smaller portions of programs that have already been created. <u>3–5. Algorithms and Programming. Program Development</u> People develop programs using an iterative process involving design, implementation, and review. Design often involves reusing existing code or remixing other programs within a community. People continuously review whether programs work as expected, and they fix, or debug, parts that do not. Repeating these steps enables people to refine and improve programs.</p>	<p><u>Level 1B</u></p> <p><u>Algorithms and Programming</u></p> <p>1B-AP-08 Compare and refine multiple algorithms for the same task and determine which is the most appropriate.</p> <p>1B-AP-10 Create programs that include sequences, events, loops, and conditionals.</p> <p>1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.</p> <p>1B-AP-12 Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.</p> <p>1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences</p> <p>1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs.</p> <p>1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.</p> <p>1B-AP-16 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development.</p> <p>1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.</p>

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
Block coding: Level 4			
Introduction to variables: Pop game	To understand how a variable can be used to keep track of the score in a game.		
Links	<p><u>Programming and development</u> Knows that users can develop their own programs, and can demonstrate this by creating a simple program in an environment that does not rely on text. (AL) Executes, checks and changes programs. (AL) Understands that programs execute by following precise instructions. (AL) Declares and assigns variables. (AB) <u>Algorithms</u> Understands what an algorithm is and is able to express simple linear (non-branching) algorithms symbolically. (AL) Detects and corrects errors, i.e. debugging in algorithms. (AL) Designs simple algorithms using loops and selection, i.e. if statements. (AL)</p>	<p><u>3–5. Algorithms and Programming. Algorithms</u> Different algorithms can achieve the same result. Some algorithms are more appropriate for a specific context than others. <u>3–5. Algorithms and Programming. Variables</u> Programming languages provide variables, which are used to store and modify data. The data type determines the values and operations that can be performed on that data. <u>3–5. Algorithms and Programming. Control</u> Control structures, including loops, event handlers, and conditionals, are used to specify the flow of execution. Conditionals selectively execute or skip instructions under different conditions. <u>3–5. Algorithms and Programming. Modularity</u> Programs can be broken down into smaller parts to facilitate their design, implementation, and review. Programs can also be created by incorporating smaller portions of programs that have already been created. <u>3–5. Algorithms and Programming. Program Development</u> People develop programs using an iterative process involving design, implementation, and review. Design often involves reusing existing code or remixing other programs within a community. People continuously review whether programs work as expected, and they fix, or debug, parts that do not. Repeating these steps enables people to refine and improve programs.</p>	<p><u>Level 1B</u> <u>Algorithms and Programming</u> 1B-AP-08 Compare and refine multiple algorithms for the same task and determine which is the most appropriate. 1B-AP-09 Create programs that use variables to store and modify data. 1B-AP-10 Create programs that include sequences, events, loops, and conditionals. 1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process. 1B-AP-12 Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features. 1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences 1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs. 1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended. 1B-AP-16 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development. 1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.</p>

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
Introduction to variables: Catch the coconuts	To use variables to keep track of the score in a game that uses conditional events.		
Links	<p><u>Programming and development</u> Knows that users can develop their own programs, and can demonstrate this by creating a simple program in an environment that does not rely on text. (AL) Executes, checks and changes programs. (AL) Understands that programs execute by following precise instructions. (AL) Declares and assigns variables. (AB) <u>Algorithms</u> Understands what an algorithm is and is able to express simple linear (non-branching) algorithms symbolically. (AL) Detects and corrects errors, i.e. debugging in algorithms. (AL) Designs simple algorithms using loops and selection, i.e. if statements. (AL)</p>	<p><u>3–5. Algorithms and Programming. Algorithms</u> Different algorithms can achieve the same result. Some algorithms are more appropriate for a specific context than others. <u>3–5. Algorithms and Programming. Variables</u> Programming languages provide variables, which are used to store and modify data. The data type determines the values and operations that can be performed on that data. <u>3–5. Algorithms and Programming. Control</u> Control structures, including loops, event handlers, and conditionals, are used to specify the flow of execution. Conditionals selectively execute or skip instructions under different conditions. <u>3–5. Algorithms and Programming. Modularity</u> Programs can be broken down into smaller parts to facilitate their design, implementation, and review. Programs can also be created by incorporating smaller portions of programs that have already been created. <u>3–5. Algorithms and Programming. Program Development</u> People develop programs using an iterative process involving design, implementation, and review. Design often involves reusing existing code or remixing other programs within a community. People continuously review whether programs work as expected, and they fix, or debug, parts that do not. Repeating these steps enables people to refine and improve programs.</p>	<p><u>Level 1B</u> <u>Algorithms and Programming</u> 1B-AP-08 Compare and refine multiple algorithms for the same task and determine which is the most appropriate. 1B-AP-09 Create programs that use variables to store and modify data. 1B-AP-10 Create programs that include sequences, events, loops, and conditionals. 1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process. 1B-AP-12 Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features. 1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences 1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs. 1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended. 1B-AP-16 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development. 1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.</p>

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
Introduction to variables: Healthy eating	To use a variable to keep track of the score in a game that uses conditional events.		
Links	<p><u>Programming and development</u> Knows that users can develop their own programs, and can demonstrate this by creating a simple program in an environment that does not rely on text. (AL) Executes, checks and changes programs. (AL) Understands that programs execute by following precise instructions. (AL) Declares and assigns variables. (AB) <u>Algorithms</u> Understands what an algorithm is and is able to express simple linear (non-branching) algorithms symbolically. (AL) Detects and corrects errors, i.e. debugging in algorithms. (AL) Designs simple algorithms using loops and selection, i.e. if statements. (AL)</p>	<p><u>3–5. Algorithms and Programming. Algorithms</u> Different algorithms can achieve the same result. Some algorithms are more appropriate for a specific context than others. <u>3–5. Algorithms and Programming. Variables</u> Programming languages provide variables, which are used to store and modify data. The data type determines the values and operations that can be performed on that data. <u>3–5. Algorithms and Programming. Control</u> Control structures, including loops, event handlers, and conditionals, are used to specify the flow of execution. Conditionals selectively execute or skip instructions under different conditions. <u>3–5. Algorithms and Programming. Modularity</u> Programs can be broken down into smaller parts to facilitate their design, implementation, and review. Programs can also be created by incorporating smaller portions of programs that have already been created. <u>3–5. Algorithms and Programming. Program Development</u> People develop programs using an iterative process involving design, implementation, and review. Design often involves reusing existing code or remixing other programs within a community. People continuously review whether programs work as expected, and they fix, or debug, parts that do not. Repeating these steps enables people to refine and improve programs.</p>	<p><u>Level 1B</u> <u>Algorithms and Programming</u> 1B-AP-08 Compare and refine multiple algorithms for the same task and determine which is the most appropriate. 1B-AP-09 Create programs that use variables to store and modify data. 1B-AP-10 Create programs that include sequences, events, loops, and conditionals. 1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process. 1B-AP-12 Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features. 1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences 1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs. 1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended. 1B-AP-16 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development. 1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.</p>

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
Introduction to variables: Tablet till	To learn how to use multiple different variables and to set the value of a variable.		
Links	<p><u>Programming and development</u> Knows that users can develop their own programs, and can demonstrate this by creating a simple program in an environment that does not rely on text. (AL) Executes, checks and changes programs. (AL) Understands that programs execute by following precise instructions. (AL) Declares and assigns variables. (AB) <u>Algorithms</u> Understands what an algorithm is and is able to express simple linear (non-branching) algorithms symbolically. (AL) Detects and corrects errors, i.e. debugging in algorithms. (AL) Designs simple algorithms using loops and selection, i.e. if statements. (AL)</p>	<p><u>3–5. Algorithms and Programming. Algorithms</u> Different algorithms can achieve the same result. Some algorithms are more appropriate for a specific context than others. <u>3–5. Algorithms and Programming. Variables</u> Programming languages provide variables, which are used to store and modify data. The data type determines the values and operations that can be performed on that data. <u>3–5. Algorithms and Programming. Control</u> Control structures, including loops, event handlers, and conditionals, are used to specify the flow of execution. Conditionals selectively execute or skip instructions under different conditions. <u>3–5. Algorithms and Programming. Modularity</u> Programs can be broken down into smaller parts to facilitate their design, implementation, and review. Programs can also be created by incorporating smaller portions of programs that have already been created. <u>3–5. Algorithms and Programming. Program Development</u> People develop programs using an iterative process involving design, implementation, and review. Design often involves reusing existing code or remixing other programs within a community. People continuously review whether programs work as expected, and they fix, or debug, parts that do not. Repeating these steps enables people to refine and improve programs.</p>	<p><u>Level 1B</u> <u>Algorithms and Programming</u> 1B-AP-08 Compare and refine multiple algorithms for the same task and determine which is the most appropriate. 1B-AP-09 Create programs that use variables to store and modify data. 1B-AP-10 Create programs that include sequences, events, loops, and conditionals. 1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process. 1B-AP-12 Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features. 1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences 1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs. 1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended. 1B-AP-16 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development. 1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.</p>

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
Introduction to variables: Pirate Gold	To use a variable to keep track of the score in a game where the score increases, decreases or resets when different conditions are met.		
Links	<p><u>Programming and development</u> Knows that users can develop their own programs, and can demonstrate this by creating a simple program in an environment that does not rely on text. (AL) Executes, checks and changes programs. (AL) Understands that programs execute by following precise instructions. (AL) Declares and assigns variables. (AB) <u>Algorithms</u> Understands what an algorithm is and is able to express simple linear (non-branching) algorithms symbolically. (AL) Detects and corrects errors, i.e. debugging in algorithms. (AL) Designs simple algorithms using loops and selection, i.e. if statements. (AL)</p>	<p><u>3–5. Algorithms and Programming. Algorithms</u> Different algorithms can achieve the same result. Some algorithms are more appropriate for a specific context than others. <u>3–5. Algorithms and Programming. Variables</u> Programming languages provide variables, which are used to store and modify data. The data type determines the values and operations that can be performed on that data. <u>3–5. Algorithms and Programming. Control</u> Control structures, including loops, event handlers, and conditionals, are used to specify the flow of execution. Conditionals selectively execute or skip instructions under different conditions. <u>3–5. Algorithms and Programming. Modularity</u> Programs can be broken down into smaller parts to facilitate their design, implementation, and review. Programs can also be created by incorporating smaller portions of programs that have already been created. <u>3–5. Algorithms and Programming. Program Development</u> People develop programs using an iterative process involving design, implementation, and review. Design often involves reusing existing code or remixing other programs within a community. People continuously review whether programs work as expected, and they fix, or debug, parts that do not. Repeating these steps enables people to refine and improve programs.</p>	<p><u>Level 1B</u> <u>Algorithms and Programming</u> 1B-AP-08 Compare and refine multiple algorithms for the same task and determine which is the most appropriate. 1B-AP-09 Create programs that use variables to store and modify data. 1B-AP-10 Create programs that include sequences, events, loops, and conditionals. 1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process. 1B-AP-12 Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features. 1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences 1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs. 1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended. 1B-AP-16 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development. 1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.</p>

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
Repetition and loops: Bugs in the garden	To use a loop to do something repeatedly in a program.		
Links	<p><u>Programming and development</u> Executes, checks and changes programs. (AL) Understands that programs execute by following precise instructions. (AL) Uses diagrams to express solutions. (AB) <u>Algorithms</u> Understands what an algorithm is and is able to express simple linear (non-branching) algorithms symbolically. (AL) Detects and corrects errors, i.e. debugging in algorithms. (AL) Designs simple algorithms using loops and selection, i.e. if statements. (AL) Designs solutions (algorithms) that use repetition and two-way selection, i.e. if, then, and else (AL)</p>	<p><u>3–5. Algorithms and Programming. Algorithms</u> Different algorithms can achieve the same result. Some algorithms are more appropriate for a specific context than others. <u>3–5. Algorithms and Programming. Variables</u> Programming languages provide variables, which are used to store and modify data. The data type determines the values and operations that can be performed on that data. <u>3–5. Algorithms and Programming. Control</u> Control structures, including loops, event handlers, and conditionals, are used to specify the flow of execution. Conditionals selectively execute or skip instructions under different conditions. <u>3–5. Algorithms and Programming. Modularity</u> Programs can be broken down into smaller parts to facilitate their design, implementation, and review. Programs can also be created by incorporating smaller portions of programs that have already been created. <u>3–5. Algorithms and Programming. Program Development</u> People develop programs using an iterative process involving design, implementation, and review. Design often involves reusing existing code or remixing other programs within a community. People continuously review whether programs work as expected, and they fix, or debug, parts that do not. Repeating these steps enables people to refine and improve programs.</p>	<p><u>Level 1B</u> <u>Algorithms and Programming</u> 1B-AP-08 Compare and refine multiple algorithms for the same task and determine which is the most appropriate. 1B-AP-09 Create programs that use variables to store and modify data. 1B-AP-10 Create programs that include sequences, events, loops, and conditionals. 1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process. 1B-AP-12 Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features. 1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences 1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs. 1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended. 1B-AP-16 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development. 1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.</p>

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
Repetition and loops: Driving me loopy	To write code that uses nested loops to create a car-driving program.		
Links	<p><u>Programming and development</u> Executes, checks and changes programs. (AL) Understands that programs execute by following precise instructions. (AL) Uses diagrams to express solutions. (AB) <u>Algorithms</u> Understands what an algorithm is and is able to express simple linear (non-branching) algorithms symbolically. (AL) Detects and corrects errors, i.e. debugging in algorithms. (AL) Designs simple algorithms using loops and selection, i.e. if statements. (AL) Designs solutions (algorithms) that use repetition and two-way selection, i.e. if, then, and else (AL)</p>	<p><u>3–5. Algorithms and Programming. Algorithms</u> Different algorithms can achieve the same result. Some algorithms are more appropriate for a specific context than others. <u>3–5. Algorithms and Programming. Variables</u> Programming languages provide variables, which are used to store and modify data. The data type determines the values and operations that can be performed on that data. <u>3–5. Algorithms and Programming. Control</u> Control structures, including loops, event handlers, and conditionals, are used to specify the flow of execution. Conditionals selectively execute or skip instructions under different conditions. <u>3–5. Algorithms and Programming. Modularity</u> Programs can be broken down into smaller parts to facilitate their design, implementation, and review. Programs can also be created by incorporating smaller portions of programs that have already been created. <u>3–5. Algorithms and Programming. Program Development</u> People develop programs using an iterative process involving design, implementation, and review. Design often involves reusing existing code or remixing other programs within a community. People continuously review whether programs work as expected, and they fix, or debug, parts that do not. Repeating these steps enables people to refine and improve programs.</p>	<p><u>Level 1B</u> <u>Algorithms and Programming</u> 1B-AP-08 Compare and refine multiple algorithms for the same task and determine which is the most appropriate. 1B-AP-09 Create programs that use variables to store and modify data. 1B-AP-10 Create programs that include sequences, events, loops, and conditionals. 1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process. 1B-AP-12 Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features. 1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences 1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs. 1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended. 1B-AP-16 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development. 1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.</p>

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
Repetition and loops: Astronaut orbit	To write the code to program a rocket to orbit round the spinning Moon, using the concepts of loops, regular or infinite repetition, and 'if statement' blocks.		
Links	<p><u>Programming and development</u> Executes, checks and changes programs. (AL) Understands that programs execute by following precise instructions. (AL) Uses diagrams to express solutions. (AB) <u>Algorithms</u> Understands what an algorithm is and is able to express simple linear (non-branching) algorithms symbolically. (AL) Detects and corrects errors, i.e. debugging in algorithms. (AL) Designs simple algorithms using loops and selection, i.e. if statements. (AL) Designs solutions (algorithms) that use repetition and two-way selection, i.e. if, then, and else (AL)</p>	<p><u>3–5. Algorithms and Programming. Algorithms</u> Different algorithms can achieve the same result. Some algorithms are more appropriate for a specific context than others. <u>3–5. Algorithms and Programming. Variables</u> Programming languages provide variables, which are used to store and modify data. The data type determines the values and operations that can be performed on that data. <u>3–5. Algorithms and Programming. Control</u> Control structures, including loops, event handlers, and conditionals, are used to specify the flow of execution. Conditionals selectively execute or skip instructions under different conditions. <u>3–5. Algorithms and Programming. Modularity</u> Programs can be broken down into smaller parts to facilitate their design, implementation, and review. Programs can also be created by incorporating smaller portions of programs that have already been created. <u>3–5. Algorithms and Programming. Program Development</u> People develop programs using an iterative process involving design, implementation, and review. Design often involves reusing existing code or remixing other programs within a community. People continuously review whether programs work as expected, and they fix, or debug, parts that do not. Repeating these steps enables people to refine and improve programs.</p>	<p><u>Level 1B</u> <u>Algorithms and Programming</u> 1B-AP-08 Compare and refine multiple algorithms for the same task and determine which is the most appropriate. 1B-AP-09 Create programs that use variables to store and modify data. 1B-AP-10 Create programs that include sequences, events, loops, and conditionals. 1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process. 1B-AP-12 Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features. 1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences 1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs. 1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended. 1B-AP-16 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development. 1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.</p>

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
Repetition and loops: Hot air balloon show	To use loops, a variable and if statements to create an animated scene of hot air balloons performing a repeating pattern in the sky.		
Links	<p><u>Programming and development</u> Executes, checks and changes programs. (AL) Understands that programs execute by following precise instructions. (AL) Uses diagrams to express solutions. (AB) <u>Algorithms</u> Understands what an algorithm is and is able to express simple linear (non-branching) algorithms symbolically. (AL) Detects and corrects errors, i.e. debugging in algorithms. (AL) Designs simple algorithms using loops and selection, i.e. if statements. (AL) Designs solutions (algorithms) that use repetition and two-way selection, i.e. if, then, and else (AL)</p>	<p><u>3–5. Algorithms and Programming. Algorithms</u> Different algorithms can achieve the same result. Some algorithms are more appropriate for a specific context than others. <u>3–5. Algorithms and Programming. Variables</u> Programming languages provide variables, which are used to store and modify data. The data type determines the values and operations that can be performed on that data. <u>3–5. Algorithms and Programming. Control</u> Control structures, including loops, event handlers, and conditionals, are used to specify the flow of execution. Conditionals selectively execute or skip instructions under different conditions. <u>3–5. Algorithms and Programming. Modularity</u> Programs can be broken down into smaller parts to facilitate their design, implementation, and review. Programs can also be created by incorporating smaller portions of programs that have already been created. <u>3–5. Algorithms and Programming. Program Development</u> People develop programs using an iterative process involving design, implementation, and review. Design often involves reusing existing code or remixing other programs within a community. People continuously review whether programs work as expected, and they fix, or debug, parts that do not. Repeating these steps enables people to refine and improve programs.</p>	<p><u>Level 1B</u> <u>Algorithms and Programming</u> 1B-AP-08 Compare and refine multiple algorithms for the same task and determine which is the most appropriate. 1B-AP-09 Create programs that use variables to store and modify data. 1B-AP-10 Create programs that include sequences, events, loops, and conditionals. 1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process. 1B-AP-12 Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features. 1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences 1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs. 1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended. 1B-AP-16 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development. 1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.</p>
Block coding: Level 5			

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
Speed, direction and coordinates: Faster and slower	To set values in code to control the speed of an object.		
Links	<p><u>Programming and development</u> Executes, checks and changes programs. (AL) Understands that programs execute by following precise instructions. (AL) Uses diagrams to express solutions. (AB) <u>Algorithms</u> Understands what an algorithm is and is able to express simple linear (non-branching) algorithms symbolically. (AL) Detects and corrects errors, i.e. debugging in algorithms. (AL) Designs simple algorithms using loops and selection, i.e. if statements. (AL) Designs solutions (algorithms) that use repetition and two-way selection, i.e. if, then, and else (AL)</p>	<p><u>3–5. Algorithms and Programming. Algorithms</u> Different algorithms can achieve the same result. Some algorithms are more appropriate for a specific context than others. <u>3–5. Algorithms and Programming. Variables</u> Programming languages provide variables, which are used to store and modify data. The data type determines the values and operations that can be performed on that data. <u>3–5. Algorithms and Programming. Control</u> Control structures, including loops, event handlers, and conditionals, are used to specify the flow of execution. Conditionals selectively execute or skip instructions under different conditions. <u>3–5. Algorithms and Programming. Modularity</u> Programs can be broken down into smaller parts to facilitate their design, implementation, and review. Programs can also be created by incorporating smaller portions of programs that have already been created. <u>3–5. Algorithms and Programming. Program Development</u> People develop programs using an iterative process involving design, implementation, and review. Design often involves reusing existing code or remixing other programs within a community. People continuously review whether programs work as expected, and they fix, or debug, parts that do not. Repeating these steps enables people to refine and improve programs.</p>	<p><u>Level 1B</u> <u>Algorithms and Programming</u> 1B-AP-08 Compare and refine multiple algorithms for the same task and determine which is the most appropriate. 1B-AP-09 Create programs that use variables to store and modify data. 1B-AP-10 Create programs that include sequences, events, loops, and conditionals. 1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process. 1B-AP-12 Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features. 1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences 1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs. 1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended. 1B-AP-16 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development. 1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.</p>

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
Speed, direction and coordinates: Speedy simulation	To use object properties (speed, heading and angle) to create a driving simulation.		
Links	<p><u>Programming and development</u> Executes, checks and changes programs. (AL) Understands that programs execute by following precise instructions. (AL) Uses diagrams to express solutions. (AB) Declares and assigns variables. (AB) <u>Algorithms</u> Understands what an algorithm is and is able to express simple linear (non-branching) algorithms symbolically. (AL) Detects and corrects errors, i.e. debugging in algorithms. (AL) Designs simple algorithms using loops and selection, i.e. if statements. (AL) Designs solutions (algorithms) that use repetition and two-way selection, i.e. if, then, and else (AL)</p>	<p><u>3–5. Algorithms and Programming. Algorithms</u> Different algorithms can achieve the same result. Some algorithms are more appropriate for a specific context than others. <u>3–5. Algorithms and Programming. Variables</u> Programming languages provide variables, which are used to store and modify data. The data type determines the values and operations that can be performed on that data. <u>3–5. Algorithms and Programming. Control</u> Control structures, including loops, event handlers, and conditionals, are used to specify the flow of execution. Conditionals selectively execute or skip instructions under different conditions. <u>3–5. Algorithms and Programming. Modularity</u> Programs can be broken down into smaller parts to facilitate their design, implementation, and review. Programs can also be created by incorporating smaller portions of programs that have already been created. <u>3–5. Algorithms and Programming. Program Development</u> People develop programs using an iterative process involving design, implementation, and review. Design often involves reusing existing code or remixing other programs within a community. People continuously review whether programs work as expected, and they fix, or debug, parts that do not. Repeating these steps enables people to refine and improve programs.</p>	<p><u>Level 1B</u> <u>Algorithms and Programming</u> 1B-AP-08 Compare and refine multiple algorithms for the same task and determine which is the most appropriate. 1B-AP-09 Create programs that use variables to store and modify data. 1B-AP-10 Create programs that include sequences, events, loops, and conditionals. 1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process. 1B-AP-12 Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features. 1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences 1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs. 1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended. 1B-AP-16 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development. 1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.</p>

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
<p>Speed, direction and coordinates: Sailing the seas</p>	<p>To create a sailing game where a boat's position on the screen is controlled by making changes to its co-ordinates.</p>		
<p>Links</p>	<p><u>Programming and development</u> Executes, checks and changes programs. (AL) Understands that programs execute by following precise instructions. (AL) Uses diagrams to express solutions. (AB) Declares and assigns variables. (AB) <u>Algorithms</u> Understands what an algorithm is and is able to express simple linear (non-branching) algorithms symbolically. (AL) Detects and corrects errors, i.e. debugging in algorithms. (AL) Designs simple algorithms using loops and selection, i.e. if statements. (AL) Designs solutions (algorithms) that use repetition and two-way selection, i.e. if, then, and else (AL)</p>	<p><u>3–5. Algorithms and Programming. Algorithms</u> Different algorithms can achieve the same result. Some algorithms are more appropriate for a specific context than others. <u>3–5. Algorithms and Programming. Variables</u> Programming languages provide variables, which are used to store and modify data. The data type determines the values and operations that can be performed on that data. <u>3–5. Algorithms and Programming. Control</u> Control structures, including loops, event handlers, and conditionals, are used to specify the flow of execution. Conditionals selectively execute or skip instructions under different conditions. <u>3–5. Algorithms and Programming. Modularity</u> Programs can be broken down into smaller parts to facilitate their design, implementation, and review. Programs can also be created by incorporating smaller portions of programs that have already been created. <u>3–5. Algorithms and Programming. Program Development</u> People develop programs using an iterative process involving design, implementation, and review. Design often involves reusing existing code or remixing other programs within a community. People continuously review whether programs work as expected, and they fix, or debug, parts that do not. Repeating these steps enables people to refine and improve programs.</p>	<p><u>Level 1B</u> <u>Algorithms and Programming</u> 1B-AP-08 Compare and refine multiple algorithms for the same task and determine which is the most appropriate. 1B-AP-09 Create programs that use variables to store and modify data. 1B-AP-10 Create programs that include sequences, events, loops, and conditionals. 1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process. 1B-AP-12 Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features. 1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences 1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs. 1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended. 1B-AP-16 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development. 1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.</p>

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
Speed, direction and coordinates: Parachuting cows	To write code including if statements to make an object rotate, and combine this with conditional events to make a game.		
Links	<p><u>Programming and development</u> Executes, checks and changes programs. (AL) Understands that programs execute by following precise instructions. (AL) Uses diagrams to express solutions. (AB) <u>Algorithms</u> Understands what an algorithm is and is able to express simple linear (non-branching) algorithms symbolically. (AL) Detects and corrects errors, i.e. debugging in algorithms. (AL) Designs simple algorithms using loops and selection, i.e. if statements. (AL) Designs solutions (algorithms) that use repetition and two-way selection, i.e. if, then, and else (AL) Declares and assigns variables. (AB)</p>	<p><u>3–5. Algorithms and Programming. Algorithms</u> Different algorithms can achieve the same result. Some algorithms are more appropriate for a specific context than others. <u>3–5. Algorithms and Programming. Variables</u> Programming languages provide variables, which are used to store and modify data. The data type determines the values and operations that can be performed on that data. <u>3–5. Algorithms and Programming. Control</u> Control structures, including loops, event handlers, and conditionals, are used to specify the flow of execution. Conditionals selectively execute or skip instructions under different conditions. <u>3–5. Algorithms and Programming. Modularity</u> Programs can be broken down into smaller parts to facilitate their design, implementation, and review. Programs can also be created by incorporating smaller portions of programs that have already been created. <u>3–5. Algorithms and Programming. Program Development</u> People develop programs using an iterative process involving design, implementation, and review. Design often involves reusing existing code or remixing other programs within a community. People continuously review whether programs work as expected, and they fix, or debug, parts that do not. Repeating these steps enables people to refine and improve programs.</p>	<p><u>Level 1B</u> <u>Algorithms and Programming</u> 1B-AP-08 Compare and refine multiple algorithms for the same task and determine which is the most appropriate. 1B-AP-09 Create programs that use variables to store and modify data. 1B-AP-10 Create programs that include sequences, events, loops, and conditionals. 1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process. 1B-AP-12 Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features. 1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences 1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs. 1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended. 1B-AP-16 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development. 1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.</p>

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
Random numbers and simulations: Racing at random	To be able to generate and display random numbers, and use these within the program for a car-racing game.		
Links	<p><u>Programming and development</u> Executes, checks and changes programs. (AL) Understands that programs execute by following precise instructions. (AL) Uses diagrams to express solutions. (AB) Declares and assigns variables. (AB)</p> <p><u>Algorithms</u> Understands what an algorithm is and is able to express simple linear (non-branching) algorithms symbolically. (AL) Detects and corrects errors, i.e. debugging in algorithms. (AL) Designs simple algorithms using loops and selection, i.e. if statements. (AL) Designs solutions (algorithms) that use repetition and two-way selection, i.e. if, then, and else (AL)</p>	<p><u>3–5. Algorithms and Programming. Algorithms</u> Different algorithms can achieve the same result. Some algorithms are more appropriate for a specific context than others.</p> <p><u>3–5. Algorithms and Programming. Variables</u> Programming languages provide variables, which are used to store and modify data. The data type determines the values and operations that can be performed on that data.</p> <p><u>3–5. Algorithms and Programming. Control</u> Control structures, including loops, event handlers, and conditionals, are used to specify the flow of execution. Conditionals selectively execute or skip instructions under different conditions.</p> <p><u>3–5. Algorithms and Programming. Modularity</u> Programs can be broken down into smaller parts to facilitate their design, implementation, and review. Programs can also be created by incorporating smaller portions of programs that have already been created.</p> <p><u>3–5. Algorithms and Programming. Program Development</u> People develop programs using an iterative process involving design, implementation, and review. Design often involves reusing existing code or remixing other programs within a community. People continuously review whether programs work as expected, and they fix, or debug, parts that do not. Repeating these steps enables people to refine and improve programs.</p>	<p><u>Level 1B</u></p> <p><u>Algorithms and Programming</u></p> <p>1B-AP-08 Compare and refine multiple algorithms for the same task and determine which is the most appropriate.</p> <p>1B-AP-09 Create programs that use variables to store and modify data.</p> <p>1B-AP-10 Create programs that include sequences, events, loops, and conditionals.</p> <p>1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.</p> <p>1B-AP-12 Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.</p> <p>1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences</p> <p>1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs.</p> <p>1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.</p> <p>1B-AP-16 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development.</p> <p>1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.</p>

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
Random numbers and simulations: Caterpillar catcher	To write code for a game that uses random numbers to move objects in different directions.		
Links	<p><u>Programming and development</u> Executes, checks and changes programs. (AL) Understands that programs execute by following precise instructions. (AL) Uses diagrams to express solutions. (AB) Declares and assigns variables. (AB) <u>Algorithms</u> Understands what an algorithm is and is able to express simple linear (non-branching) algorithms symbolically. (AL) Detects and corrects errors, i.e. debugging in algorithms. (AL) Designs simple algorithms using loops and selection, i.e. if statements. (AL) Designs solutions (algorithms) that use repetition and two-way selection, i.e. if, then, and else (AL)</p>	<p><u>3–5. Algorithms and Programming. Algorithms</u> Different algorithms can achieve the same result. Some algorithms are more appropriate for a specific context than others. <u>3–5. Algorithms and Programming. Variables</u> Programming languages provide variables, which are used to store and modify data. The data type determines the values and operations that can be performed on that data. <u>3–5. Algorithms and Programming. Control</u> Control structures, including loops, event handlers, and conditionals, are used to specify the flow of execution. Conditionals selectively execute or skip instructions under different conditions. <u>3–5. Algorithms and Programming. Modularity</u> Programs can be broken down into smaller parts to facilitate their design, implementation, and review. Programs can also be created by incorporating smaller portions of programs that have already been created. <u>3–5. Algorithms and Programming. Program Development</u> People develop programs using an iterative process involving design, implementation, and review. Design often involves reusing existing code or remixing other programs within a community. People continuously review whether programs work as expected, and they fix, or debug, parts that do not. Repeating these steps enables people to refine and improve programs.</p>	<p><u>Level 1B</u> <u>Algorithms and Programming</u> 1B-AP-08 Compare and refine multiple algorithms for the same task and determine which is the most appropriate. 1B-AP-09 Create programs that use variables to store and modify data. 1B-AP-10 Create programs that include sequences, events, loops, and conditionals. 1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process. 1B-AP-12 Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features. 1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences 1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs. 1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended. 1B-AP-16 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development. 1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.</p>

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
Random numbers and simulations: Cross the road	To write code that uses random numbers to move objects at random speeds and headings, and use this to create a game.		
Links	<p><u>Programming and development</u> Executes, checks and changes programs. (AL) Understands that programs execute by following precise instructions. (AL) Uses diagrams to express solutions. (AB) Declares and assigns variables. (AB) <u>Algorithms</u> Understands what an algorithm is and is able to express simple linear (non-branching) algorithms symbolically. (AL) Detects and corrects errors, i.e. debugging in algorithms. (AL) Designs simple algorithms using loops and selection, i.e. if statements. (AL) Designs solutions (algorithms) that use repetition and two-way selection, i.e. if, then, and else (AL)</p>	<p><u>3–5. Algorithms and Programming. Algorithms</u> Different algorithms can achieve the same result. Some algorithms are more appropriate for a specific context than others. <u>3–5. Algorithms and Programming. Variables</u> Programming languages provide variables, which are used to store and modify data. The data type determines the values and operations that can be performed on that data. <u>3–5. Algorithms and Programming. Control</u> Control structures, including loops, event handlers, and conditionals, are used to specify the flow of execution. Conditionals selectively execute or skip instructions under different conditions. <u>3–5. Algorithms and Programming. Modularity</u> Programs can be broken down into smaller parts to facilitate their design, implementation, and review. Programs can also be created by incorporating smaller portions of programs that have already been created. <u>3–5. Algorithms and Programming. Program Development</u> People develop programs using an iterative process involving design, implementation, and review. Design often involves reusing existing code or remixing other programs within a community. People continuously review whether programs work as expected, and they fix, or debug, parts that do not. Repeating these steps enables people to refine and improve programs.</p>	<p><u>Level 1B</u> <u>Algorithms and Programming</u> 1B-AP-08 Compare and refine multiple algorithms for the same task and determine which is the most appropriate. 1B-AP-09 Create programs that use variables to store and modify data. 1B-AP-10 Create programs that include sequences, events, loops, and conditionals. 1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process. 1B-AP-12 Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features. 1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences 1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs. 1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended. 1B-AP-16 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development. 1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.</p>

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
Random numbers and simulations: Ping Pong	To create a ping-pong game, using random headings in specific ranges.		
Links	<p><u>Programming and development</u> Executes, checks and changes programs. (AL) Understands that programs execute by following precise instructions. (AL) Uses diagrams to express solutions. (AB) Declares and assigns variables. (AB)</p> <p><u>Algorithms</u> Understands what an algorithm is and is able to express simple linear (non-branching) algorithms symbolically. (AL) Detects and corrects errors, i.e. debugging in algorithms. (AL) Designs simple algorithms using loops and selection, i.e. if statements. (AL) Designs solutions (algorithms) that use repetition and two-way selection, i.e. if, then, and else (AL)</p>	<p><u>3–5. Algorithms and Programming. Algorithms</u> Different algorithms can achieve the same result. Some algorithms are more appropriate for a specific context than others.</p> <p><u>3–5. Algorithms and Programming. Variables</u> Programming languages provide variables, which are used to store and modify data. The data type determines the values and operations that can be performed on that data.</p> <p><u>3–5. Algorithms and Programming. Control</u> Control structures, including loops, event handlers, and conditionals, are used to specify the flow of execution. Conditionals selectively execute or skip instructions under different conditions.</p> <p><u>3–5. Algorithms and Programming. Modularity</u> Programs can be broken down into smaller parts to facilitate their design, implementation, and review. Programs can also be created by incorporating smaller portions of programs that have already been created.</p> <p><u>3–5. Algorithms and Programming. Program Development</u> People develop programs using an iterative process involving design, implementation, and review. Design often involves reusing existing code or remixing other programs within a community. People continuously review whether programs work as expected, and they fix, or debug, parts that do not. Repeating these steps enables people to refine and improve programs.</p>	<p><u>Level 1B</u></p> <p><u>Algorithms and Programming</u></p> <p>1B-AP-08 Compare and refine multiple algorithms for the same task and determine which is the most appropriate.</p> <p>1B-AP-09 Create programs that use variables to store and modify data.</p> <p>1B-AP-10 Create programs that include sequences, events, loops, and conditionals.</p> <p>1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.</p> <p>1B-AP-12 Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.</p> <p>1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences</p> <p>1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs.</p> <p>1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.</p> <p>1B-AP-16 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development.</p> <p>1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.</p>

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
Random numbers and simulations: Pinball	To use random numbers in combination with variables and conditional hit events to create a realistic pinball game.		
Links	<p><u>Programming and development</u> Executes, checks and changes programs. (AL) Understands that programs execute by following precise instructions. (AL) Uses diagrams to express solutions. (AB) Declares and assigns variables. (AB)</p> <p><u>Algorithms</u> Understands what an algorithm is and is able to express simple linear (non-branching) algorithms symbolically. (AL) Detects and corrects errors, i.e. debugging in algorithms. (AL) Designs simple algorithms using loops and selection, i.e. if statements. (AL) Designs solutions (algorithms) that use repetition and two-way selection, i.e. if, then, and else (AL)</p>	<p><u>3–5. Algorithms and Programming. Algorithms</u> Different algorithms can achieve the same result. Some algorithms are more appropriate for a specific context than others.</p> <p><u>3–5. Algorithms and Programming. Variables</u> Programming languages provide variables, which are used to store and modify data. The data type determines the values and operations that can be performed on that data.</p> <p><u>3–5. Algorithms and Programming. Control</u> Control structures, including loops, event handlers, and conditionals, are used to specify the flow of execution. Conditionals selectively execute or skip instructions under different conditions.</p> <p><u>3–5. Algorithms and Programming. Modularity</u> Programs can be broken down into smaller parts to facilitate their design, implementation, and review. Programs can also be created by incorporating smaller portions of programs that have already been created.</p> <p><u>3–5. Algorithms and Programming. Program Development</u> People develop programs using an iterative process involving design, implementation, and review. Design often involves reusing existing code or remixing other programs within a community. People continuously review whether programs work as expected, and they fix, or debug, parts that do not. Repeating these steps enables people to refine and improve programs.</p>	<p><u>Level 1B</u></p> <p><u>Algorithms and Programming</u></p> <p>1B-AP-08 Compare and refine multiple algorithms for the same task and determine which is the most appropriate.</p> <p>1B-AP-09 Create programs that use variables to store and modify data.</p> <p>1B-AP-10 Create programs that include sequences, events, loops, and conditionals.</p> <p>1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.</p> <p>1B-AP-12 Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.</p> <p>1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences</p> <p>1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs.</p> <p>1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.</p> <p>1B-AP-16 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development.</p> <p>1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.</p>
Block coding: Level 6			

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
More complex variables: Shape-shifting	To write code that prompts the user to input the value of a variable, and use this to create an interactive block chart.		
Links	<p><u>Programming and development</u> Executes, checks and changes programs. (AL) Understands that programs execute by following precise instructions. (AL) Uses diagrams to express solutions. (AB) Declares and assigns variables. (AB) <u>Algorithms</u> Understands what an algorithm is and is able to express simple linear (non-branching) algorithms symbolically. (AL) Detects and corrects errors, i.e. debugging in algorithms. (AL) Designs simple algorithms using loops and selection, i.e. if statements. (AL) Designs solutions (algorithms) that use repetition and two-way selection, i.e. if, then, and else. (AL)</p>	<p><u>6–8. Algorithms and Programming. Algorithms</u> Algorithms affect how people interact with computers and the way computers respond. People design algorithms that are generalizable to many situations. Algorithms that are readable are easier to follow, test, and debug. <u>6–8. Algorithms and Programming. Variables</u> Programmers create variables to store data values of selected types. A meaningful identifier is assigned to each variable to access and perform operations on the value by name. Variables enable the flexibility to represent different situations, process different sets of data, and produce varying outputs. <u>6–8. Algorithms and Programming. Control</u> Programmers select and combine control structures, such as loops, event handlers, and conditionals, to create more complex program behavior. <u>6–8. Algorithms and Programming. Modularity</u> Programs use procedures to organize code, hide implementation details, and make code easier to reuse. Procedures can be repurposed in new programs. Defining parameters for procedures can generalize behavior and increase reusability. <u>6–8. Algorithms and Programming. Program Development</u> People design meaningful solutions for others by defining a problem’s criteria and constraints, carefully considering the diverse needs and wants of the community, and testing whether criteria and constraints were met.</p>	<p><u>Algorithms and Programming</u></p> <p>Level 1B 1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process. 1B-AP-12 Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features. 1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences 1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs. 1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended. 1B-AP-16 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development. 1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.</p> <p>Level 2 2-AP-11 Create clearly named variables that represent different data types and perform operations on their values. 2-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.</p>

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
More complex variables: Pop challenge	To use my knowledge of variables to make a balloon pop game that gets harder as users score more points.		
Links	<p><u>Programming and development</u> Executes, checks and changes programs. (AL) Understands that programs execute by following precise instructions. (AL) Uses diagrams to express solutions. (AB) Declares and assigns variables. (AB) Uses post-tested loop, e.g. 'until', and a sequence of selection statements in programs, including an if, then and else statement. (AL) <u>Algorithms</u> Understands what an algorithm is and is able to express simple linear (non-branching) algorithms symbolically. (AL) Detects and corrects errors, i.e. debugging in algorithms. (AL) Designs simple algorithms using loops and selection, i.e. if statements. (AL) Designs solutions (algorithms) that use repetition and two-way selection, i.e. if, then, and else. (AL)</p>	<p><u>6–8. Algorithms and Programming. Algorithms</u> Algorithms affect how people interact with computers and the way computers respond. People design algorithms that are generalizable to many situations. Algorithms that are readable are easier to follow, test, and debug. <u>6–8. Algorithms and Programming. Variables</u> Programmers create variables to store data values of selected types. A meaningful identifier is assigned to each variable to access and perform operations on the value by name. Variables enable the flexibility to represent different situations, process different sets of data, and produce varying outputs. <u>6–8. Algorithms and Programming. Control</u> Programmers select and combine control structures, such as loops, event handlers, and conditionals, to create more complex program behavior. <u>6–8. Algorithms and Programming. Modularity</u> Programs use procedures to organize code, hide implementation details, and make code easier to reuse. Procedures can be repurposed in new programs. Defining parameters for procedures can generalize behavior and increase reusability. <u>6–8. Algorithms and Programming. Program Development</u> People design meaningful solutions for others by defining a problem’s criteria and constraints, carefully considering the diverse needs and wants of the community, and testing whether criteria and constraints were met.</p>	<p><u>Algorithms and Programming</u></p> <p>Level 1B 1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process. 1B-AP-12 Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features. 1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences 1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs. 1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended. 1B-AP-16 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development. 1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.</p> <p>Level 2 2-AP-11 Create clearly named variables that represent different data types and perform operations on their values. 2-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.</p>

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
<p>More complex variables: Toyshop till</p>	<p>To write the code for a shopping till using variables to store and calculate values.</p>		
<p>Links</p>	<p><u>Programming and development</u> Executes, checks and changes programs. (AL) Understands that programs execute by following precise instructions. (AL) Uses diagrams to express solutions. (AB) Declares and assigns variables. (AB) <u>Algorithms</u> Understands what an algorithm is and is able to express simple linear (non-branching) algorithms symbolically. (AL) Detects and corrects errors, i.e. debugging in algorithms. (AL) Designs simple algorithms using loops and selection, i.e. if statements. (AL) Designs solutions (algorithms) that use repetition and two-way selection, i.e. if, then, and else. (AL)</p>	<p><u>6–8. Algorithms and Programming. Algorithms</u> Algorithms affect how people interact with computers and the way computers respond. People design algorithms that are generalizable to many situations. Algorithms that are readable are easier to follow, test, and debug. <u>6–8. Algorithms and Programming. Variables</u> Programmers create variables to store data values of selected types. A meaningful identifier is assigned to each variable to access and perform operations on the value by name. Variables enable the flexibility to represent different situations, process different sets of data, and produce varying outputs. <u>6–8. Algorithms and Programming. Control</u> Programmers select and combine control structures, such as loops, event handlers, and conditionals, to create more complex program behavior. <u>6–8. Algorithms and Programming. Modularity</u> Programs use procedures to organize code, hide implementation details, and make code easier to reuse. Procedures can be repurposed in new programs. Defining parameters for procedures can generalize behavior and increase reusability. <u>6–8. Algorithms and Programming. Program Development</u> People design meaningful solutions for others by defining a problem’s criteria and constraints, carefully considering the diverse needs and wants of the community, and testing whether criteria and constraints were met.</p>	<p><u>Algorithms and Programming</u></p> <p>Level 1B 1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process. 1B-AP-12 Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features. 1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences 1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs. 1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended. 1B-AP-16 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development. 1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.</p> <p>Level 2 2-AP-11 Create clearly named variables that represent different data types and perform operations on their values. 2-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.</p>

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
<p>More complex variables: Stopwatch</p>	<p>To create a stopwatch with stop, start, and reset buttons, and both digital and analogue displays.</p>		
<p>Links</p>	<p><u>Programming and development</u> Executes, checks and changes programs. (AL) Understands that programs execute by following precise instructions. (AL) Uses diagrams to express solutions. (AB) Declares and assigns variables. (AB) Uses post-tested loop, e.g. 'until', and a sequence of selection statements in programs, including an if, then and else statement. (AL) <u>Algorithms</u> Understands what an algorithm is and is able to express simple linear (non-branching) algorithms symbolically. (AL) Detects and corrects errors, i.e. debugging in algorithms. (AL) Designs simple algorithms using loops and selection, i.e. if statements. (AL) Designs solutions (algorithms) that use repetition and two-way selection, i.e. if, then, and else. (AL)</p>	<p><u>6–8. Algorithms and Programming. Algorithms</u> Algorithms affect how people interact with computers and the way computers respond. People design algorithms that are generalizable to many situations. Algorithms that are readable are easier to follow, test, and debug. <u>6–8. Algorithms and Programming. Variables</u> Programmers create variables to store data values of selected types. A meaningful identifier is assigned to each variable to access and perform operations on the value by name. Variables enable the flexibility to represent different situations, process different sets of data, and produce varying outputs. <u>6–8. Algorithms and Programming. Control</u> Programmers select and combine control structures, such as loops, event handlers, and conditionals, to create more complex program behavior. <u>6–8. Algorithms and Programming. Modularity</u> Programs use procedures to organize code, hide implementation details, and make code easier to reuse. Procedures can be repurposed in new programs. Defining parameters for procedures can generalize behavior and increase reusability. <u>6–8. Algorithms and Programming. Program Development</u> People design meaningful solutions for others by defining a problem's criteria and constraints, carefully considering the diverse needs and wants of the community, and testing whether criteria and constraints were met.</p>	<p><u>Algorithms and Programming</u></p> <p>Level 1B 1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process. 1B-AP-12 Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features. 1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences 1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs. 1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended. 1B-AP-16 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development. 1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.</p> <p>Level 2 2-AP-11 Create clearly named variables that represent different data types and perform operations on their values. 2-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.</p>

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
Object properties: Don't feed the birds	To create a game where players stop objects moving by changing their properties.		
Links	<p><u>Programming and development</u> Executes, checks and changes programs. (AL) Understands that programs execute by following precise instructions. (AL) Declares and assigns variables. (AB)</p> <p><u>Algorithms</u> Understands what an algorithm is and is able to express simple linear (non-branching) algorithms symbolically. (AL) Detects and corrects errors, i.e. debugging in algorithms. (AL) Designs simple algorithms using loops and selection, i.e. if statements. (AL) Designs solutions (algorithms) that use repetition and two-way selection, i.e. if, then, and else. (AL) Uses diagrams to express solutions. (AB)</p>	<p><u>6–8. Algorithms and Programming. Algorithms</u> Algorithms affect how people interact with computers and the way computers respond. People design algorithms that are generalizable to many situations. Algorithms that are readable are easier to follow, test, and debug.</p> <p><u>6–8. Algorithms and Programming. Variables</u> Programmers create variables to store data values of selected types. A meaningful identifier is assigned to each variable to access and perform operations on the value by name. Variables enable the flexibility to represent different situations, process different sets of data, and produce varying outputs.</p> <p><u>6–8. Algorithms and Programming. Control</u> Programmers select and combine control structures, such as loops, event handlers, and conditionals, to create more complex program behavior.</p> <p><u>6–8. Algorithms and Programming. Modularity</u> Programs use procedures to organize code, hide implementation details, and make code easier to reuse. Procedures can be repurposed in new programs. Defining parameters for procedures can generalize behavior and increase reusability.</p> <p><u>6–8. Algorithms and Programming. Program Development</u> People design meaningful solutions for others by defining a problem's criteria and constraints, carefully considering the diverse needs and wants of the community, and testing whether criteria and constraints were met.</p>	<p><u>Algorithms and Programming</u></p> <p>Level 1B</p> <p>1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.</p> <p>1B-AP-12 Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.</p> <p>1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences</p> <p>1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs.</p> <p>1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.</p> <p>1B-AP-16 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development.</p> <p>1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.</p> <p>Level 2</p> <p>2-AP-11 Create clearly named variables that represent different data types and perform operations on their values.</p> <p>2-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.</p>

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
Object properties: Rocket blaster	To write code that detects the properties of an object and passes the value of these properties (or a set of parameters) to other objects, and to use this to create a space game.		
Links	<p><u>Programming and development</u> Executes, checks and changes programs. (AL) Understands that programs execute by following precise instructions. (AL) Declares and assigns variables. (AB)</p> <p><u>Algorithms</u> Understands what an algorithm is and is able to express simple linear (non-branching) algorithms symbolically. (AL) Detects and corrects errors, i.e. debugging in algorithms. (AL) Designs simple algorithms using loops and selection, i.e. if statements. (AL) Designs solutions (algorithms) that use repetition and two-way selection, i.e. if, then, and else. (AL) Uses diagrams to express solutions. (AB)</p>	<p><u>6–8. Algorithms and Programming. Algorithms</u> Algorithms affect how people interact with computers and the way computers respond. People design algorithms that are generalizable to many situations. Algorithms that are readable are easier to follow, test, and debug.</p> <p><u>6–8. Algorithms and Programming. Variables</u> Programmers create variables to store data values of selected types. A meaningful identifier is assigned to each variable to access and perform operations on the value by name. Variables enable the flexibility to represent different situations, process different sets of data, and produce varying outputs.</p> <p><u>6–8. Algorithms and Programming. Control</u> Programmers select and combine control structures, such as loops, event handlers, and conditionals, to create more complex program behavior.</p> <p><u>6–8. Algorithms and Programming. Modularity</u> Programs use procedures to organize code, hide implementation details, and make code easier to reuse. Procedures can be repurposed in new programs. Defining parameters for procedures can generalize behavior and increase reusability.</p> <p><u>6–8. Algorithms and Programming. Program Development</u> People design meaningful solutions for others by defining a problem’s criteria and constraints, carefully considering the diverse needs and wants of the community, and testing whether criteria and constraints were met.</p>	<p><u>Algorithms and Programming</u></p> <p>Level 1B 1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process. 1B-AP-12 Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.</p> <p>1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences</p> <p>1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs.</p> <p>1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.</p> <p>1B-AP-16 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development.</p> <p>1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.</p> <p>Level 2 2-AP-11 Create clearly named variables that represent different data types and perform operations on their values.</p> <p>2-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.</p>

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
Object properties: Football fun	To make a football game that passes the speed and heading of the pointer's movement to a ball on the screen.		
Links	<p><u>Programming and development</u> Executes, checks and changes programs. (AL) Understands that programs execute by following precise instructions. (AL) Declares and assigns variables. (AB)</p> <p><u>Algorithms</u> Understands what an algorithm is and is able to express simple linear (non-branching) algorithms symbolically. (AL) Detects and corrects errors, i.e. debugging in algorithms. (AL) Designs simple algorithms using loops and selection, i.e. if statements. (AL) Designs solutions (algorithms) that use repetition and two-way selection, i.e. if, then, and else. (AL) Uses diagrams to express solutions. (AB)</p>	<p><u>6–8. Algorithms and Programming. Algorithms</u> Algorithms affect how people interact with computers and the way computers respond. People design algorithms that are generalizable to many situations. Algorithms that are readable are easier to follow, test, and debug.</p> <p><u>6–8. Algorithms and Programming. Variables</u> Programmers create variables to store data values of selected types. A meaningful identifier is assigned to each variable to access and perform operations on the value by name. Variables enable the flexibility to represent different situations, process different sets of data, and produce varying outputs.</p> <p><u>6–8. Algorithms and Programming. Control</u> Programmers select and combine control structures, such as loops, event handlers, and conditionals, to create more complex program behavior.</p> <p><u>6–8. Algorithms and Programming. Modularity</u> Programs use procedures to organize code, hide implementation details, and make code easier to reuse. Procedures can be repurposed in new programs. Defining parameters for procedures can generalize behavior and increase reusability.</p> <p><u>6–8. Algorithms and Programming. Program Development</u> People design meaningful solutions for others by defining a problem's criteria and constraints, carefully considering the diverse needs and wants of the community, and testing whether criteria and constraints were met.</p>	<p><u>Algorithms and Programming</u></p> <p>Level 1B 1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process. 1B-AP-12 Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.</p> <p>1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences</p> <p>1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs.</p> <p>1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.</p> <p>1B-AP-16 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development.</p> <p>1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.</p> <p>Level 2 2-AP-11 Create clearly named variables that represent different data types and perform operations on their values.</p> <p>2-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.</p>

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
Object properties: Sheepdog	To make a game that moves objects around by getting information from events and passing object properties. To learn how to pass properties from one object to a second in order to make the second object move relative to the first.		
Links	<p><u>Programming and development</u> Executes, checks and changes programs. (AL) Understands that programs execute by following precise instructions. (AL) Declares and assigns variables. (AB)</p> <p><u>Algorithms</u> Understands what an algorithm is and is able to express simple linear (non-branching) algorithms symbolically. (AL) Detects and corrects errors, i.e. debugging in algorithms. (AL) Designs simple algorithms using loops and selection, i.e. if statements. (AL) Designs solutions (algorithms) that use repetition and two-way selection, i.e. if, then, and else. (AL) Uses diagrams to express solutions. (AB)</p>	<p><u>6–8. Algorithms and Programming. Algorithms</u> Algorithms affect how people interact with computers and the way computers respond. People design algorithms that are generalizable to many situations. Algorithms that are readable are easier to follow, test, and debug.</p> <p><u>6–8. Algorithms and Programming. Variables</u> Programmers create variables to store data values of selected types. A meaningful identifier is assigned to each variable to access and perform operations on the value by name. Variables enable the flexibility to represent different situations, process different sets of data, and produce varying outputs.</p> <p><u>6–8. Algorithms and Programming. Control</u> Programmers select and combine control structures, such as loops, event handlers, and conditionals, to create more complex program behavior.</p> <p><u>6–8. Algorithms and Programming. Modularity</u> Programs use procedures to organize code, hide implementation details, and make code easier to reuse. Procedures can be repurposed in new programs. Defining parameters for procedures can generalize behavior and increase reusability.</p> <p><u>6–8. Algorithms and Programming. Program Development</u> People design meaningful solutions for others by defining a problem’s criteria and constraints, carefully considering the diverse needs and wants of the community, and testing whether criteria and constraints were met.</p>	<p><u>Algorithms and Programming</u></p> <p>Level 1B 1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process. 1B-AP-12 Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.</p> <p>1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences</p> <p>1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs.</p> <p>1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.</p> <p>1B-AP-16 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development.</p> <p>1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.</p> <p>Level 2 2-AP-11 Create clearly named variables that represent different data types and perform operations on their values.</p> <p>2-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.</p>

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
Object properties: Golf game	To create a golf game by writing code that accesses and uses object properties, including passing the value of these properties to other objects (passing a set of parameters).		
Links	<p><u>Programming and development</u> Executes, checks and changes programs. (AL) Understands that programs execute by following precise instructions. (AL) Declares and assigns variables. (AB)</p> <p><u>Algorithms</u> Understands what an algorithm is and is able to express simple linear (non-branching) algorithms symbolically. (AL) Detects and corrects errors, i.e. debugging in algorithms. (AL) Designs simple algorithms using loops and selection, i.e. if statements. (AL) Designs solutions (algorithms) that use repetition and two-way selection, i.e. if, then, and else. (AL) Uses diagrams to express solutions. (AB)</p>	<p><u>6–8. Algorithms and Programming. Algorithms</u> Algorithms affect how people interact with computers and the way computers respond. People design algorithms that are generalizable to many situations. Algorithms that are readable are easier to follow, test, and debug.</p> <p><u>6–8. Algorithms and Programming. Variables</u> Programmers create variables to store data values of selected types. A meaningful identifier is assigned to each variable to access and perform operations on the value by name. Variables enable the flexibility to represent different situations, process different sets of data, and produce varying outputs.</p> <p><u>6–8. Algorithms and Programming. Control</u> Programmers select and combine control structures, such as loops, event handlers, and conditionals, to create more complex program behavior.</p> <p><u>6–8. Algorithms and Programming. Modularity</u> Programs use procedures to organize code, hide implementation details, and make code easier to reuse. Procedures can be repurposed in new programs. Defining parameters for procedures can generalize behavior and increase reusability.</p> <p><u>6–8. Algorithms and Programming. Program Development</u> People design meaningful solutions for others by defining a problem’s criteria and constraints, carefully considering the diverse needs and wants of the community, and testing whether criteria and constraints were met.</p>	<p><u>Algorithms and Programming</u></p> <p>Level 1B 1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process. 1B-AP-12 Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.</p> <p>1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences</p> <p>1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs.</p> <p>1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.</p> <p>1B-AP-16 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development.</p> <p>1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.</p> <p>Level 2 2-AP-11 Create clearly named variables that represent different data types and perform operations on their values.</p> <p>2-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.</p>
Python: Levels 5-6			

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
Introduction to Python: Python printing	To learn to input information in Python and generate an output by running the code.		
Links		<p><u>3–5. Algorithms and Programming. Algorithms</u> Different algorithms can achieve the same result. Some algorithms are more appropriate for a specific context than others.</p> <p><u>3–5. Algorithms and Programming. Variables</u> Programming languages provide variables, which are used to store and modify data. The data type determines the values and operations that can be performed on that data.</p> <p><u>3–5. Algorithms and Programming. Control</u> Control structures, including loops, event handlers, and conditionals, are used to specify the flow of execution. Conditionals selectively execute or skip instructions under different conditions.</p> <p><u>3–5. Algorithms and Programming. Modularity</u> Programs can be broken down into smaller parts to facilitate their design, implementation, and review. Programs can also be created by incorporating smaller portions of programs that have already been created.</p> <p><u>3–5. Algorithms and Programming. Program Development</u> People develop programs using an iterative process involving design, implementation, and review. Design often involves reusing existing code or remixing other programs within a community. People continuously review whether programs work as expected, and they fix, or debug, parts that do not. Repeating these steps enables people to refine and improve programs.</p>	<p><u>Level 1B</u></p> <p><u>Algorithms and Programming</u></p> <p>1B-AP-08 Compare and refine multiple algorithms for the same task and determine which is the most appropriate.</p> <p>1B-AP-10 Create programs that include sequences, events, loops, and conditionals.</p> <p>1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.</p> <p>1B-AP-12 Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.</p> <p>1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences</p> <p>1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs.</p> <p>1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.</p> <p>1B-AP-16 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development.</p> <p>1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.</p>
Introduction to Python: Simple calculations	To learn how to use Python to make simple calculations and recognise symbols for multiplication (*) and division (/).		

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
Links		<p><u>3–5. Algorithms and Programming. Algorithms</u> Different algorithms can achieve the same result. Some algorithms are more appropriate for a specific context than others.</p> <p><u>3–5. Algorithms and Programming. Variables</u> Programming languages provide variables, which are used to store and modify data. The data type determines the values and operations that can be performed on that data.</p> <p><u>3–5. Algorithms and Programming. Control</u> Control structures, including loops, event handlers, and conditionals, are used to specify the flow of execution. Conditionals selectively execute or skip instructions under different conditions.</p> <p><u>3–5. Algorithms and Programming. Modularity</u> Programs can be broken down into smaller parts to facilitate their design, implementation, and review. Programs can also be created by incorporating smaller portions of programs that have already been created.</p> <p><u>3–5. Algorithms and Programming. Program Development</u> People develop programs using an iterative process involving design, implementation, and review. Design often involves reusing existing code or remixing other programs within a community. People continuously review whether programs work as expected, and they fix, or debug, parts that do not. Repeating these steps enables people to refine and improve programs.</p>	<p><u>Level 1B</u></p> <p><u>Algorithms and Programming</u></p> <p>1B-AP-08 Compare and refine multiple algorithms for the same task and determine which is the most appropriate.</p> <p>1B-AP-10 Create programs that include sequences, events, loops, and conditionals.</p> <p>1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.</p> <p>1B-AP-12 Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.</p> <p>1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences</p> <p>1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs.</p> <p>1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.</p> <p>1B-AP-16 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development.</p> <p>1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.</p>
Introduction to Python: Inputs and variables	To learn how inputs and variables work.		
Links		<u>3–5. Algorithms and Programming. Algorithms</u>	<u>Level 1B</u>

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
		<p>Different algorithms can achieve the same result. Some algorithms are more appropriate for a specific context than others.</p> <p><u>3–5. Algorithms and Programming. Variables</u> Programming languages provide variables, which are used to store and modify data. The data type determines the values and operations that can be performed on that data.</p> <p><u>3–5. Algorithms and Programming. Control</u> Control structures, including loops, event handlers, and conditionals, are used to specify the flow of execution. Conditionals selectively execute or skip instructions under different conditions.</p> <p><u>3–5. Algorithms and Programming. Modularity</u> Programs can be broken down into smaller parts to facilitate their design, implementation, and review. Programs can also be created by incorporating smaller portions of programs that have already been created.</p> <p><u>3–5. Algorithms and Programming. Program Development</u> People develop programs using an iterative process involving design, implementation, and review. Design often involves reusing existing code or remixing other programs within a community. People continuously review whether programs work as expected, and they fix, or debug, parts that do not. Repeating these steps enables people to refine and improve programs.</p>	<p><u>Algorithms and Programming</u></p> <p>1B-AP-08 Compare and refine multiple algorithms for the same task and determine which is the most appropriate.</p> <p>1B-AP-09 Create programs that use variables to store and modify data.</p> <p>1B-AP-10 Create programs that include sequences, events, loops, and conditionals.</p> <p>1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.</p> <p>1B-AP-12 Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.</p> <p>1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences</p> <p>1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs.</p> <p>1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.</p> <p>1B-AP-16 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development.</p> <p>1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.</p>
<p>Introduction to Python: Selection and input</p>	<p>To learn how to use selection and inputs.</p>		
<p>Links</p>		<p><u>3–5. Algorithms and Programming. Algorithms</u> Different algorithms can achieve the same result. Some algorithms are more appropriate for a specific context than others.</p>	<p><u>Level 1B</u></p> <p><u>Algorithms and Programming</u></p>

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
		<p><u>3–5. Algorithms and Programming. Variables</u> Programming languages provide variables, which are used to store and modify data. The data type determines the values and operations that can be performed on that data.</p> <p><u>3–5. Algorithms and Programming. Control</u> Control structures, including loops, event handlers, and conditionals, are used to specify the flow of execution. Conditionals selectively execute or skip instructions under different conditions.</p> <p><u>3–5. Algorithms and Programming. Modularity</u> Programs can be broken down into smaller parts to facilitate their design, implementation, and review. Programs can also be created by incorporating smaller portions of programs that have already been created.</p> <p><u>3–5. Algorithms and Programming. Program Development</u> People develop programs using an iterative process involving design, implementation, and review. Design often involves reusing existing code or remixing other programs within a community. People continuously review whether programs work as expected, and they fix, or debug, parts that do not. Repeating these steps enables people to refine and improve programs.</p>	<p>1B-AP-08 Compare and refine multiple algorithms for the same task and determine which is the most appropriate.</p> <p>1B-AP-09 Create programs that use variables to store and modify data.</p> <p>1B-AP-10 Create programs that include sequences, events, loops, and conditionals.</p> <p>1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.</p> <p>1B-AP-12 Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.</p> <p>1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences</p> <p>1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs.</p> <p>1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.</p> <p>1B-AP-16 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development.</p> <p>1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.</p>
Introduction to Python: Calculations with variables	To learn how to use variables to store values.		
Links		<p><u>3–5. Algorithms and Programming. Algorithms</u> Different algorithms can achieve the same result. Some algorithms are more appropriate for a specific context than others.</p>	<p><u>Level 1B</u> <u>Algorithms and Programming</u></p>

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
		<p><u>3–5. Algorithms and Programming. Variables</u> Programming languages provide variables, which are used to store and modify data. The data type determines the values and operations that can be performed on that data.</p> <p><u>3–5. Algorithms and Programming. Control</u> Control structures, including loops, event handlers, and conditionals, are used to specify the flow of execution. Conditionals selectively execute or skip instructions under different conditions.</p> <p><u>3–5. Algorithms and Programming. Modularity</u> Programs can be broken down into smaller parts to facilitate their design, implementation, and review. Programs can also be created by incorporating smaller portions of programs that have already been created.</p> <p><u>3–5. Algorithms and Programming. Program Development</u> People develop programs using an iterative process involving design, implementation, and review. Design often involves reusing existing code or remixing other programs within a community. People continuously review whether programs work as expected, and they fix, or debug, parts that do not. Repeating these steps enables people to refine and improve programs.</p>	<p>1B-AP-08 Compare and refine multiple algorithms for the same task and determine which is the most appropriate.</p> <p>1B-AP-09 Create programs that use variables to store and modify data.</p> <p>1B-AP-10 Create programs that include sequences, events, loops, and conditionals.</p> <p>1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.</p> <p>1B-AP-12 Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.</p> <p>1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences</p> <p>1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs.</p> <p>1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.</p> <p>1B-AP-16 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development.</p> <p>1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.</p>
Introduction to Python: Quiz with variables	To learn how to use the 'if' command and the 'input' with variables to make a quiz which keeps score.		
Links		<p><u>3–5. Algorithms and Programming. Algorithms</u> Different algorithms can achieve the same result. Some algorithms are more appropriate for a specific context than others.</p> <p><u>3–5. Algorithms and Programming. Variables</u></p>	<p><u>Level 1B</u></p> <p><u>Algorithms and Programming</u></p>

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
		<p>Programming languages provide variables, which are used to store and modify data. The data type determines the values and operations that can be performed on that data.</p> <p><u>3-5. Algorithms and Programming. Control</u> Control structures, including loops, event handlers, and conditionals, are used to specify the flow of execution. Conditionals selectively execute or skip instructions under different conditions.</p> <p><u>3-5. Algorithms and Programming. Modularity</u> Programs can be broken down into smaller parts to facilitate their design, implementation, and review. Programs can also be created by incorporating smaller portions of programs that have already been created.</p> <p><u>3-5. Algorithms and Programming. Program Development</u> People develop programs using an iterative process involving design, implementation, and review. Design often involves reusing existing code or remixing other programs within a community. People continuously review whether programs work as expected, and they fix, or debug, parts that do not. Repeating these steps enables people to refine and improve programs.</p>	<p>1B-AP-08 Compare and refine multiple algorithms for the same task and determine which is the most appropriate.</p> <p>1B-AP-09 Create programs that use variables to store and modify data.</p> <p>1B-AP-10 Create programs that include sequences, events, loops, and conditionals.</p> <p>1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.</p> <p>1B-AP-12 Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.</p> <p>1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences</p> <p>1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs.</p> <p>1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.</p> <p>1B-AP-16 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development.</p> <p>1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.</p>
Python graphics: Introducing python graphics	To understand what turtle graphics are and how to code lines and shapes using the turtle.		
Links		<p><u>3-5. Algorithms and Programming. Algorithms</u> Different algorithms can achieve the same result. Some algorithms are more appropriate for a specific context than others.</p> <p><u>3-5. Algorithms and Programming. Variables</u></p>	<p><u>Level 1B</u></p> <p><u>Algorithms and Programming</u></p>

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
		<p>Programming languages provide variables, which are used to store and modify data. The data type determines the values and operations that can be performed on that data.</p> <p><u>3–5. Algorithms and Programming. Control</u> Control structures, including loops, event handlers, and conditionals, are used to specify the flow of execution. Conditionals selectively execute or skip instructions under different conditions.</p> <p><u>3–5. Algorithms and Programming. Modularity</u> Programs can be broken down into smaller parts to facilitate their design, implementation, and review. Programs can also be created by incorporating smaller portions of programs that have already been created.</p> <p><u>3–5. Algorithms and Programming. Program Development</u> People develop programs using an iterative process involving design, implementation, and review. Design often involves reusing existing code or remixing other programs within a community. People continuously review whether programs work as expected, and they fix, or debug, parts that do not. Repeating these steps enables people to refine and improve programs.</p>	<p>1B-AP-08 Compare and refine multiple algorithms for the same task and determine which is the most appropriate.</p> <p>1B-AP-09 Create programs that use variables to store and modify data.</p> <p>1B-AP-10 Create programs that include sequences, events, loops, and conditionals.</p> <p>1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.</p> <p>1B-AP-12 Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.</p> <p>1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences</p> <p>1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs.</p> <p>1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.</p> <p>1B-AP-16 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development.</p> <p>1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.</p> <p><u>Level 2</u> 2-AP-14 Create procedures with parameters to organize code and make it easier to reuse. 2-AP-16 Incorporate existing code, media, and libraries into original programs, and give attribution.</p>

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
<p>Python graphics: Circles and colour</p>	<p>To learn how to use the turtle to draw circles and change their colour.</p>	<p><u>3–5. Algorithms and Programming. Algorithms</u> Different algorithms can achieve the same result. Some algorithms are more appropriate for a specific context than others.</p> <p><u>3–5. Algorithms and Programming. Variables</u> Programming languages provide variables, which are used to store and modify data. The data type determines the values and operations that can be performed on that data.</p> <p><u>3–5. Algorithms and Programming. Control</u> Control structures, including loops, event handlers, and conditionals, are used to specify the flow of execution. Conditionals selectively execute or skip instructions under different conditions.</p> <p><u>3–5. Algorithms and Programming. Modularity</u> Programs can be broken down into smaller parts to facilitate their design, implementation, and review. Programs can also be created by incorporating smaller portions of programs that have already been created.</p> <p><u>3–5. Algorithms and Programming. Program Development</u> People develop programs using an iterative process involving design, implementation, and review. Design often involves reusing existing code or remixing other programs within a community. People continuously review whether programs work as expected, and they fix, or debug, parts that do not. Repeating these steps enables people to refine and improve programs.</p>	<p><u>Level 1B</u></p> <p><u>Algorithms and Programming</u></p> <p>1B-AP-08 Compare and refine multiple algorithms for the same task and determine which is the most appropriate.</p> <p>1B-AP-09 Create programs that use variables to store and modify data.</p> <p>1B-AP-10 Create programs that include sequences, events, loops, and conditionals.</p> <p>1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.</p> <p>1B-AP-12 Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.</p> <p>1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences</p> <p>1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs.</p> <p>1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.</p> <p>1B-AP-16 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development.</p> <p>1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.</p> <p><u>Level 2</u></p> <p>2-AP-14 Create procedures with parameters to organize code and make it easier to reuse.</p>

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
			2-AP-16 Incorporate existing code, media, and libraries into original programs, and give attribution.
Links			
Python graphics: Loops and shapes	To learn how to use loops to draw shapes in Python.		
Links		<p><u>3–5. Algorithms and Programming. Algorithms</u> Different algorithms can achieve the same result. Some algorithms are more appropriate for a specific context than others.</p> <p><u>3–5. Algorithms and Programming. Variables</u> Programming languages provide variables, which are used to store and modify data. The data type determines the values and operations that can be performed on that data.</p> <p><u>3–5. Algorithms and Programming. Control</u> Control structures, including loops, event handlers, and conditionals, are used to specify the flow of execution. Conditionals selectively execute or skip instructions under different conditions.</p> <p><u>3–5. Algorithms and Programming. Modularity</u> Programs can be broken down into smaller parts to facilitate their design, implementation, and review. Programs can also be created by incorporating smaller portions of programs that have already been created.</p> <p><u>3–5. Algorithms and Programming. Program Development</u> People develop programs using an iterative process involving design, implementation, and review. Design often involves reusing existing code or remixing other programs within a community. People continuously review whether programs work as expected, and they fix, or debug, parts that do not. Repeating these steps enables people to refine and improve programs.</p>	<p><u>Level 1B</u></p> <p><u>Algorithms and Programming</u></p> <p>1B-AP-08 Compare and refine multiple algorithms for the same task and determine which is the most appropriate.</p> <p>1B-AP-09 Create programs that use variables to store and modify data.</p> <p>1B-AP-10 Create programs that include sequences, events, loops, and conditionals.</p> <p>1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.</p> <p>1B-AP-12 Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.</p> <p>1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences</p> <p>1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs.</p> <p>1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.</p>

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
			<p>1B-AP-16 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development.</p> <p>1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.</p> <p><u>Level 2</u></p> <p>2-AP-14 Create procedures with parameters to organize code and make it easier to reuse.</p> <p>2-AP-16 Incorporate existing code, media, and libraries into original programs, and give attribution.</p>
<p>Python graphics: Simple pictures</p>	<p>To learn how to program the turtle to create a simple face using coordinates.</p>		
<p>Links</p>		<p><u>3–5. Algorithms and Programming. Algorithms</u> Different algorithms can achieve the same result. Some algorithms are more appropriate for a specific context than others.</p> <p><u>3–5. Algorithms and Programming. Variables</u> Programming languages provide variables, which are used to store and modify data. The data type determines the values and operations that can be performed on that data.</p> <p><u>3–5. Algorithms and Programming. Control</u> Control structures, including loops, event handlers, and conditionals, are used to specify the flow of execution. Conditionals selectively execute or skip instructions under different conditions.</p> <p><u>3–5. Algorithms and Programming. Modularity</u> Programs can be broken down into smaller parts to facilitate their design, implementation, and review. Programs can also be created by incorporating smaller portions of programs that have already been created.</p> <p><u>3–5. Algorithms and Programming. Program Development</u> People develop programs using an iterative process involving design, implementation, and review. Design often involves reusing existing code or remixing other programs within a community. People continuously review whether programs work as expected, and they fix, or debug, parts that do not. Repeating these steps enables people to refine and improve programs.</p>	<p><u>Level 1B</u></p> <p><u>Algorithms and Programming</u></p> <p>1B-AP-08 Compare and refine multiple algorithms for the same task and determine which is the most appropriate.</p> <p>1B-AP-09 Create programs that use variables to store and modify data.</p> <p>1B-AP-10 Create programs that include sequences, events, loops, and conditionals.</p> <p>1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.</p> <p>1B-AP-12 Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.</p> <p>1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences</p>

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
			<p>1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs.</p> <p>1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.</p> <p>1B-AP-16 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development.</p> <p>1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.</p> <p><u>Level 2</u></p> <p>2-AP-14 Create procedures with parameters to organize code and make it easier to reuse.</p> <p>2-AP-16 Incorporate existing code, media, and libraries into original programs, and give attribution.</p>
<p>Python graphics: Line size and backgrounds</p>	<p>To develop the use of turtle graphics to affect background colour, line and width.</p>		
<p>Links</p>	<p><u>3-5. Algorithms and Programming. Algorithms</u> Different algorithms can achieve the same result. Some algorithms are more appropriate for a specific context than others.</p> <p><u>3-5. Algorithms and Programming. Variables</u> Programming languages provide variables, which are used to store and modify data. The data type determines the values and operations that can be performed on that data.</p> <p><u>3-5. Algorithms and Programming. Control</u> Control structures, including loops, event handlers, and conditionals, are used to specify the flow of execution. Conditionals selectively execute or skip instructions under different conditions.</p> <p><u>3-5. Algorithms and Programming. Modularity</u> Programs can be broken down into smaller parts to facilitate their design, implementation, and review. Programs can also be created by incorporating smaller portions of programs that have already been created.</p> <p><u>3-5. Algorithms and Programming. Program Development</u></p>		<p><u>Level 1B</u></p> <p><u>Algorithms and Programming</u></p> <p>1B-AP-08 Compare and refine multiple algorithms for the same task and determine which is the most appropriate.</p> <p>1B-AP-09 Create programs that use variables to store and modify data.</p> <p>1B-AP-10 Create programs that include sequences, events, loops, and conditionals.</p> <p>1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.</p> <p>1B-AP-12 Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.</p>

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
		<p>People develop programs using an iterative process involving design, implementation, and review. Design often involves reusing existing code or remixing other programs within a community. People continuously review whether programs work as expected, and they fix, or debug, parts that do not. Repeating these steps enables people to refine and improve programs.</p>	<p>1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences</p> <p>1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs.</p> <p>1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.</p> <p>1B-AP-16 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development.</p> <p>1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.</p> <p><u>Level 2</u></p> <p>2-AP-14 Create procedures with parameters to organize code and make it easier to reuse.</p> <p>2-AP-16 Incorporate existing code, media, and libraries into original programs, and give attribution.</p>
Python graphics: Loops	To learn how to use loops to create different images.		
Links		<p><u>3–5. Algorithms and Programming. Algorithms</u> Different algorithms can achieve the same result. Some algorithms are more appropriate for a specific context than others.</p> <p><u>3–5. Algorithms and Programming. Variables</u> Programming languages provide variables, which are used to store and modify data. The data type determines the values and operations that can be performed on that data.</p> <p><u>3–5. Algorithms and Programming. Control</u> Control structures, including loops, event handlers, and conditionals, are used to specify the flow of execution. Conditionals selectively execute or skip instructions under different conditions.</p> <p><u>3–5. Algorithms and Programming. Modularity</u></p>	<p><u>Level 1B</u></p> <p><u>Algorithms and Programming</u></p> <p>1B-AP-08 Compare and refine multiple algorithms for the same task and determine which is the most appropriate.</p> <p>1B-AP-09 Create programs that use variables to store and modify data.</p> <p>1B-AP-10 Create programs that include sequences, events, loops, and conditionals.</p>

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
		<p>Programs can be broken down into smaller parts to facilitate their design, implementation, and review. Programs can also be created by incorporating smaller portions of programs that have already been created.</p> <p><u>3–5. Algorithms and Programming. Program Development</u> People develop programs using an iterative process involving design, implementation, and review. Design often involves reusing existing code or remixing other programs within a community. People continuously review whether programs work as expected, and they fix, or debug, parts that do not. Repeating these steps enables people to refine and improve programs.</p>	<p>1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.</p> <p>1B-AP-12 Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.</p> <p>1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences</p> <p>1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs.</p> <p>1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.</p> <p>1B-AP-16 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development.</p> <p>1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.</p> <p><u>Level 2</u> 2-AP-14 Create procedures with parameters to organize code and make it easier to reuse. 2-AP-16 Incorporate existing code, media, and libraries into original programs, and give attribution.</p>
<p>Random numbers and simulations: Random numbers</p>	<p>To learn how to use the random numbers library in Python.</p>		
<p>Links</p>		<p><u>6–8. Algorithms and Programming. Algorithms</u> Algorithms affect how people interact with computers and the way computers respond. People design algorithms that are generalizable to many situations. Algorithms that are readable are easier to follow, test, and debug.</p> <p><u>6–8. Algorithms and Programming. Variables</u></p>	<p><u>Level 1B</u> <u>Algorithms and Programming</u></p> <p>1B-AP-08 Compare and refine multiple algorithms for the same task and determine which is the most appropriate.</p>

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
		<p>Programmers create variables to store data values of selected types. A meaningful identifier is assigned to each variable to access and perform operations on the value by name. Variables enable the flexibility to represent different situations, process different sets of data, and produce varying outputs.</p> <p><u>6–8. Algorithms and Programming. Control</u> Programmers select and combine control structures, such as loops, event handlers, and conditionals, to create more complex program behavior.</p> <p><u>6–8. Algorithms and Programming. Modularity</u> Programs use procedures to organize code, hide implementation details, and make code easier to reuse. Procedures can be repurposed in new programs. Defining parameters for procedures can generalize behavior and increase reusability.</p> <p><u>6–8. Algorithms and Programming. Program Development</u> People design meaningful solutions for others by defining a problem’s criteria and constraints, carefully considering the diverse needs and wants of the community, and testing whether criteria and constraints were met.</p>	<p>1B-AP-09 Create programs that use variables to store and modify data.</p> <p>1B-AP-10 Create programs that include sequences, events, loops, and conditionals.</p> <p>1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.</p> <p>1B-AP-12 Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.</p> <p>1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences</p> <p>1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs.</p> <p>1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.</p> <p>1B-AP-16 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development.</p> <p>1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.</p> <p><u>Level 2</u> 2-AP-14 Create procedures with parameters to organize code and make it easier to reuse. 2-AP-16 Incorporate existing code, media, and libraries into original programs, and give attribution.</p>
<p>Random numbers and simulations: Weather forecast</p>	<p>To learn how to combine random numbers and text in Python using different commands.</p>		
<p>Links</p>		<p><u>6–8. Algorithms and Programming. Algorithms</u></p>	<p><u>Level 1B</u></p>

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
		<p>Algorithms affect how people interact with computers and the way computers respond. People design algorithms that are generalizable to many situations. Algorithms that are readable are easier to follow, test, and debug.</p> <p><u>6–8. Algorithms and Programming. Variables</u> Programmers create variables to store data values of selected types. A meaningful identifier is assigned to each variable to access and perform operations on the value by name. Variables enable the flexibility to represent different situations, process different sets of data, and produce varying outputs.</p> <p><u>6–8. Algorithms and Programming. Control</u> Programmers select and combine control structures, such as loops, event handlers, and conditionals, to create more complex program behavior.</p> <p><u>6–8. Algorithms and Programming. Modularity</u> Programs use procedures to organize code, hide implementation details, and make code easier to reuse. Procedures can be repurposed in new programs. Defining parameters for procedures can generalize behavior and increase reusability.</p> <p><u>6–8. Algorithms and Programming. Program Development</u> People design meaningful solutions for others by defining a problem’s criteria and constraints, carefully considering the diverse needs and wants of the community, and testing whether criteria and constraints were met.</p>	<p><u>Algorithms and Programming</u></p> <p>1B-AP-08 Compare and refine multiple algorithms for the same task and determine which is the most appropriate.</p> <p>1B-AP-09 Create programs that use variables to store and modify data.</p> <p>1B-AP-10 Create programs that include sequences, events, loops, and conditionals.</p> <p>1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.</p> <p>1B-AP-12 Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.</p> <p>1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences</p> <p>1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs.</p> <p>1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.</p> <p>1B-AP-16 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development.</p> <p>1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.</p> <p><u>Level 2</u></p> <p>2-AP-14 Create procedures with parameters to organize code and make it easier to reuse.</p> <p>2-AP-16 Incorporate existing code, media, and libraries into original programs, and give attribution.</p>

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
Random numbers and simulations: Robotic poetry	To learn how arrays can be used to produce phrases within Python.		Commented [FL(1): Note: Arrays are not mentioned in US Framework. They are mentioned in Computer Standards, but not until grades 9-12.
Links		<p><u>6–8. Algorithms and Programming. Algorithms</u> Algorithms affect how people interact with computers and the way computers respond. People design algorithms that are generalizable to many situations. Algorithms that are readable are easier to follow, test, and debug.</p> <p><u>6–8. Algorithms and Programming. Variables</u> Programmers create variables to store data values of selected types. A meaningful identifier is assigned to each variable to access and perform operations on the value by name. Variables enable the flexibility to represent different situations, process different sets of data, and produce varying outputs.</p> <p><u>6–8. Algorithms and Programming. Control</u> Programmers select and combine control structures, such as loops, event handlers, and conditionals, to create more complex program behavior.</p> <p><u>6–8. Algorithms and Programming. Modularity</u> Programs use procedures to organize code, hide implementation details, and make code easier to reuse. Procedures can be repurposed in new programs. Defining parameters for procedures can generalize behavior and increase reusability.</p> <p><u>6–8. Algorithms and Programming. Program Development</u> People design meaningful solutions for others by defining a problem’s criteria and constraints, carefully considering the diverse needs and wants of the community, and testing whether criteria and constraints were met.</p>	<p><u>Level 1B</u></p> <p><u>Algorithms and Programming</u></p> <p>1B-AP-08 Compare and refine multiple algorithms for the same task and determine which is the most appropriate.</p> <p>1B-AP-09 Create programs that use variables to store and modify data.</p> <p>1B-AP-10 Create programs that include sequences, events, loops, and conditionals.</p> <p>1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.</p> <p>1B-AP-12 Modify, remix, or incorporate portions of an existing program into one’s own work, to develop something new or add more advanced features.</p> <p>1B-AP-13 Use an iterative process to plan the development of a program by including others’ perspectives and considering user preferences</p> <p>1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs.</p> <p>1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.</p> <p>1B-AP-16 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development.</p> <p>1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.</p> <p><u>Level 2</u></p>

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
			<p>2-AP-14 Create procedures with parameters to organize code and make it easier to reuse.</p> <p>2-AP-16 Incorporate existing code, media, and libraries into original programs, and give attribution.</p>
<p>Random numbers and simulations: Once (or twice) upon a time</p>	<p>To use text and arrays in Python to randomise elements of a story.</p>		
<p>Links</p>		<p><u>6–8. Algorithms and Programming. Algorithms</u> Algorithms affect how people interact with computers and the way computers respond. People design algorithms that are generalizable to many situations. Algorithms that are readable are easier to follow, test, and debug.</p> <p><u>6–8. Algorithms and Programming. Variables</u> Programmers create variables to store data values of selected types. A meaningful identifier is assigned to each variable to access and perform operations on the value by name. Variables enable the flexibility to represent different situations, process different sets of data, and produce varying outputs.</p> <p><u>6–8. Algorithms and Programming. Control</u> Programmers select and combine control structures, such as loops, event handlers, and conditionals, to create more complex program behavior.</p> <p><u>6–8. Algorithms and Programming. Modularity</u> Programs use procedures to organize code, hide implementation details, and make code easier to reuse. Procedures can be repurposed in new programs. Defining parameters for procedures can generalize behavior and increase reusability.</p> <p><u>6–8. Algorithms and Programming. Program Development</u> People design meaningful solutions for others by defining a problem’s criteria and constraints, carefully considering the diverse needs and wants of the community, and testing whether criteria and constraints were met.</p>	<p><u>Level 1B</u></p> <p><u>Algorithms and Programming</u></p> <p>1B-AP-08 Compare and refine multiple algorithms for the same task and determine which is the most appropriate.</p> <p>1B-AP-09 Create programs that use variables to store and modify data.</p> <p>1B-AP-10 Create programs that include sequences, events, loops, and conditionals.</p> <p>1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.</p> <p>1B-AP-12 Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.</p> <p>1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences</p> <p>1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs.</p> <p>1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.</p> <p>1B-AP-16 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development.</p>

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
			<p>1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.</p> <p><u>Level 2</u></p> <p>2-AP-14 Create procedures with parameters to organize code and make it easier to reuse.</p> <p>2-AP-16 Incorporate existing code, media, and libraries into original programs, and give attribution.</p>
<p>Random numbers and simulations: Random colours</p>	<p>To learn how to combine the random library and the graphics library to create a variety of effects.</p>		
<p>Links</p>		<p><u>6–8. Algorithms and Programming. Algorithms</u> Algorithms affect how people interact with computers and the way computers respond. People design algorithms that are generalizable to many situations. Algorithms that are readable are easier to follow, test, and debug.</p> <p><u>6–8. Algorithms and Programming. Variables</u> Programmers create variables to store data values of selected types. A meaningful identifier is assigned to each variable to access and perform operations on the value by name. Variables enable the flexibility to represent different situations, process different sets of data, and produce varying outputs.</p> <p><u>6–8. Algorithms and Programming. Control</u> Programmers select and combine control structures, such as loops, event handlers, and conditionals, to create more complex program behavior.</p> <p><u>6–8. Algorithms and Programming. Modularity</u> Programs use procedures to organize code, hide implementation details, and make code easier to reuse. Procedures can be repurposed in new programs. Defining parameters for procedures can generalize behavior and increase reusability.</p> <p><u>6–8. Algorithms and Programming. Program Development</u> People design meaningful solutions for others by defining a problem’s criteria and constraints, carefully considering the diverse needs and wants of the community, and testing whether criteria and constraints were met.</p>	<p><u>Level 1B</u></p> <p><u>Algorithms and Programming</u></p> <p>1B-AP-08 Compare and refine multiple algorithms for the same task and determine which is the most appropriate.</p> <p>1B-AP-09 Create programs that use variables to store and modify data.</p> <p>1B-AP-10 Create programs that include sequences, events, loops, and conditionals.</p> <p>1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.</p> <p>1B-AP-12 Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.</p> <p>1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences</p> <p>1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs.</p>

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
			<p>1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.</p> <p>1B-AP-16 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development.</p> <p>1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.</p> <p><u>Level 2</u></p> <p>2-AP-14 Create procedures with parameters to organize code and make it easier to reuse.</p> <p>2-AP-16 Incorporate existing code, media, and libraries into original programs, and give attribution.</p>
<p>Random numbers and simulations: Random shapes</p>	<p>To learn how to combine loops with random numbers and graphics libraries to create 2D shapes.</p>		
<p>Links</p>		<p><u>6–8. Algorithms and Programming. Algorithms</u> Algorithms affect how people interact with computers and the way computers respond. People design algorithms that are generalizable to many situations. Algorithms that are readable are easier to follow, test, and debug.</p> <p><u>6–8. Algorithms and Programming. Variables</u> Programmers create variables to store data values of selected types. A meaningful identifier is assigned to each variable to access and perform operations on the value by name. Variables enable the flexibility to represent different situations, process different sets of data, and produce varying outputs.</p> <p><u>6–8. Algorithms and Programming. Control</u> Programmers select and combine control structures, such as loops, event handlers, and conditionals, to create more complex program behavior.</p> <p><u>6–8. Algorithms and Programming. Modularity</u> Programs use procedures to organize code, hide implementation details, and make code easier to reuse. Procedures can be repurposed in new programs. Defining parameters for procedures can generalize behavior and increase reusability.</p> <p><u>6–8. Algorithms and Programming. Program Development</u></p>	<p><u>Level 1B</u></p> <p><u>Algorithms and Programming</u></p> <p>1B-AP-08 Compare and refine multiple algorithms for the same task and determine which is the most appropriate.</p> <p>1B-AP-09 Create programs that use variables to store and modify data.</p> <p>1B-AP-10 Create programs that include sequences, events, loops, and conditionals.</p> <p>1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.</p> <p>1B-AP-12 Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.</p>

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
		<p>People design meaningful solutions for others by defining a problem’s criteria and constraints, carefully considering the diverse needs and wants of the community, and testing whether criteria and constraints were met.</p>	<p>1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences</p> <p>1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs.</p> <p>1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.</p> <p>1B-AP-16 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development.</p> <p>1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.</p> <p><u>Level 2</u></p> <p>2-AP-14 Create procedures with parameters to organize code and make it easier to reuse.</p> <p>2-AP-16 Incorporate existing code, media, and libraries into original programs, and give attribution.</p>
<p>Python functions: Introduction to functions</p>	<p>To learn how to define and call a function which uses parameters.</p>	<p><u>6–8. Algorithms and Programming. Algorithms</u> Algorithms affect how people interact with computers and the way computers respond. People design algorithms that are generalizable to many situations. Algorithms that are readable are easier to follow, test, and debug.</p> <p><u>6–8. Algorithms and Programming. Variables</u> Programmers create variables to store data values of selected types. A meaningful identifier is assigned to each variable to access and perform operations on the value by name. Variables enable the flexibility to represent different situations, process different sets of data, and produce varying outputs.</p> <p><u>6–8. Algorithms and Programming. Control</u></p>	<p><u>Level 1B</u></p> <p><u>Algorithms and Programming</u></p> <p>1B-AP-08 Compare and refine multiple algorithms for the same task and determine which is the most appropriate.</p> <p>1B-AP-09 Create programs that use variables to store and modify data.</p> <p>1B-AP-10 Create programs that include sequences, events, loops, and conditionals.</p>
<p>Links</p>			

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
		<p>Programmers select and combine control structures, such as loops, event handlers, and conditionals, to create more complex program behavior.</p> <p><u>6–8. Algorithms and Programming. Modularity</u> Programs use procedures to organize code, hide implementation details, and make code easier to reuse. Procedures can be repurposed in new programs. Defining parameters for procedures can generalize behavior and increase reusability.</p> <p><u>6–8. Algorithms and Programming. Program Development</u> People design meaningful solutions for others by defining a problem’s criteria and constraints, carefully considering the diverse needs and wants of the community, and testing whether criteria and constraints were met.</p>	<p>1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.</p> <p>1B-AP-12 Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.</p> <p>1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences</p> <p>1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs.</p> <p>1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.</p> <p>1B-AP-16 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development.</p> <p>1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.</p> <p><u>Level 2</u> 2-AP-14 Create procedures with parameters to organize code and make it easier to reuse. 2-AP-16 Incorporate existing code, media, and libraries into original programs, and give attribution.</p>
<p>Python functions: Drawing graphics with functions</p>	<p>To learn how to define and call a function which uses parameters to create turtle graphics.</p>		
<p>Links</p>		<p><u>6–8. Algorithms and Programming. Algorithms</u> Algorithms affect how people interact with computers and the way computers respond. People design algorithms that are generalizable to many situations. Algorithms that are readable are easier to follow, test, and debug.</p> <p><u>6–8. Algorithms and Programming. Variables</u></p>	<p><u>Level 1B</u> <u>Algorithms and Programming</u></p> <p>1B-AP-08 Compare and refine multiple algorithms for the same task and determine which is the most appropriate.</p>

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
		<p>Programmers create variables to store data values of selected types. A meaningful identifier is assigned to each variable to access and perform operations on the value by name. Variables enable the flexibility to represent different situations, process different sets of data, and produce varying outputs.</p> <p><u>6–8. Algorithms and Programming. Control</u> Programmers select and combine control structures, such as loops, event handlers, and conditionals, to create more complex program behavior.</p> <p><u>6–8. Algorithms and Programming. Modularity</u> Programs use procedures to organize code, hide implementation details, and make code easier to reuse. Procedures can be repurposed in new programs. Defining parameters for procedures can generalize behavior and increase reusability.</p> <p><u>6–8. Algorithms and Programming. Program Development</u> People design meaningful solutions for others by defining a problem’s criteria and constraints, carefully considering the diverse needs and wants of the community, and testing whether criteria and constraints were met.</p>	<p>1B-AP-09 Create programs that use variables to store and modify data.</p> <p>1B-AP-10 Create programs that include sequences, events, loops, and conditionals.</p> <p>1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.</p> <p>1B-AP-12 Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.</p> <p>1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences</p> <p>1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs.</p> <p>1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.</p> <p>1B-AP-16 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development.</p> <p>1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.</p> <p><u>Level 2</u> 2-AP-14 Create procedures with parameters to organize code and make it easier to reuse. 2-AP-16 Incorporate existing code, media, and libraries into original programs, and give attribution.</p>
<p>Python functions: Drawing trees with functions</p>	<p>To learn how to draw a forest scene with the turtle, using random numbers and a loop.</p>		
<p>Links</p>		<p><u>6–8. Algorithms and Programming. Algorithms</u></p>	<p><u>Level 1B</u></p>

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
		<p>Algorithms affect how people interact with computers and the way computers respond. People design algorithms that are generalizable to many situations. Algorithms that are readable are easier to follow, test, and debug.</p> <p><u>6–8. Algorithms and Programming. Variables</u> Programmers create variables to store data values of selected types. A meaningful identifier is assigned to each variable to access and perform operations on the value by name. Variables enable the flexibility to represent different situations, process different sets of data, and produce varying outputs.</p> <p><u>6–8. Algorithms and Programming. Control</u> Programmers select and combine control structures, such as loops, event handlers, and conditionals, to create more complex program behavior.</p> <p><u>6–8. Algorithms and Programming. Modularity</u> Programs use procedures to organize code, hide implementation details, and make code easier to reuse. Procedures can be repurposed in new programs. Defining parameters for procedures can generalize behavior and increase reusability.</p> <p><u>6–8. Algorithms and Programming. Program Development</u> People design meaningful solutions for others by defining a problem’s criteria and constraints, carefully considering the diverse needs and wants of the community, and testing whether criteria and constraints were met.</p>	<p><u>Algorithms and Programming</u></p> <p>1B-AP-08 Compare and refine multiple algorithms for the same task and determine which is the most appropriate.</p> <p>1B-AP-09 Create programs that use variables to store and modify data.</p> <p>1B-AP-10 Create programs that include sequences, events, loops, and conditionals.</p> <p>1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.</p> <p>1B-AP-12 Modify, remix, or incorporate portions of an existing program into one’s own work, to develop something new or add more advanced features.</p> <p>1B-AP-13 Use an iterative process to plan the development of a program by including others’ perspectives and considering user preferences</p> <p>1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs.</p> <p>1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.</p> <p>1B-AP-16 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development.</p> <p>1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.</p> <p><u>Level 2</u></p> <p>2-AP-14 Create procedures with parameters to organize code and make it easier to reuse.</p> <p>2-AP-16 Incorporate existing code, media, and libraries into original programs, and give attribution.</p>

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
Python functions: Drawing bear faces with functions	To learn how to use the turtle to create a set of bear faces which are different sizes and in different positions.		
Links		<p><u>6–8. Algorithms and Programming. Algorithms</u> Algorithms affect how people interact with computers and the way computers respond. People design algorithms that are generalizable to many situations. Algorithms that are readable are easier to follow, test, and debug.</p> <p><u>6–8. Algorithms and Programming. Variables</u> Programmers create variables to store data values of selected types. A meaningful identifier is assigned to each variable to access and perform operations on the value by name. Variables enable the flexibility to represent different situations, process different sets of data, and produce varying outputs.</p> <p><u>6–8. Algorithms and Programming. Control</u> Programmers select and combine control structures, such as loops, event handlers, and conditionals, to create more complex program behavior.</p> <p><u>6–8. Algorithms and Programming. Modularity</u> Programs use procedures to organize code, hide implementation details, and make code easier to reuse. Procedures can be repurposed in new programs. Defining parameters for procedures can generalize behavior and increase reusability.</p> <p><u>6–8. Algorithms and Programming. Program Development</u> People design meaningful solutions for others by defining a problem’s criteria and constraints, carefully considering the diverse needs and wants of the community, and testing whether criteria and constraints were met.</p>	<p><u>Level 1B</u></p> <p><u>Algorithms and Programming</u></p> <p>1B-AP-08 Compare and refine multiple algorithms for the same task and determine which is the most appropriate.</p> <p>1B-AP-09 Create programs that use variables to store and modify data.</p> <p>1B-AP-10 Create programs that include sequences, events, loops, and conditionals.</p> <p>1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.</p> <p>1B-AP-12 Modify, remix, or incorporate portions of an existing program into one’s own work, to develop something new or add more advanced features.</p> <p>1B-AP-13 Use an iterative process to plan the development of a program by including others’ perspectives and considering user preferences</p> <p>1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs.</p> <p>1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.</p> <p>1B-AP-16 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development.</p> <p>1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.</p> <p><u>Level 2</u></p>

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
			<p>2-AP-14 Create procedures with parameters to organize code and make it easier to reuse.</p> <p>2-AP-16 Incorporate existing code, media, and libraries into original programs, and give attribution.</p>
<p>Python functions: Drawing emoticons with functions</p>	<p>To learn how to use the turtle to create a random set of emoticons with different expressions.</p>		
<p>Links</p>	<p><u>6–8. Algorithms and Programming. Algorithms</u> Algorithms affect how people interact with computers and the way computers respond. People design algorithms that are generalizable to many situations. Algorithms that are readable are easier to follow, test, and debug.</p> <p><u>6–8. Algorithms and Programming. Variables</u> Programmers create variables to store data values of selected types. A meaningful identifier is assigned to each variable to access and perform operations on the value by name. Variables enable the flexibility to represent different situations, process different sets of data, and produce varying outputs.</p> <p><u>6–8. Algorithms and Programming. Control</u> Programmers select and combine control structures, such as loops, event handlers, and conditionals, to create more complex program behavior.</p> <p><u>6–8. Algorithms and Programming. Modularity</u> Programs use procedures to organize code, hide implementation details, and make code easier to reuse. Procedures can be repurposed in new programs. Defining parameters for procedures can generalize behavior and increase reusability.</p> <p><u>6–8. Algorithms and Programming. Program Development</u> People design meaningful solutions for others by defining a problem’s criteria and constraints, carefully considering the diverse needs and wants of the community, and testing whether criteria and constraints were met.</p>		<p><u>Level 1B</u></p> <p><u>Algorithms and Programming</u></p> <p>1B-AP-08 Compare and refine multiple algorithms for the same task and determine which is the most appropriate.</p> <p>1B-AP-09 Create programs that use variables to store and modify data.</p> <p>1B-AP-10 Create programs that include sequences, events, loops, and conditionals.</p> <p>1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.</p> <p>1B-AP-12 Modify, remix, or incorporate portions of an existing program into one’s own work, to develop something new or add more advanced features.</p> <p>1B-AP-13 Use an iterative process to plan the development of a program by including others’ perspectives and considering user preferences</p> <p>1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs.</p> <p>1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.</p> <p>1B-AP-16 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development.</p>

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
			<p>1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.</p> <p><u>Level 2</u></p> <p>2-AP-14 Create procedures with parameters to organize code and make it easier to reuse.</p> <p>2-AP-16 Incorporate existing code, media, and libraries into original programs, and give attribution.</p>
<p>Python functions: Drawing snowflakes with functions</p>	<p>To learn how to use the turtle to create a random set of snowflakes in different positions on the screen.</p>		
<p>Links</p>		<p><u>6–8. Algorithms and Programming. Algorithms</u> Algorithms affect how people interact with computers and the way computers respond. People design algorithms that are generalizable to many situations. Algorithms that are readable are easier to follow, test, and debug.</p> <p><u>6–8. Algorithms and Programming. Variables</u> Programmers create variables to store data values of selected types. A meaningful identifier is assigned to each variable to access and perform operations on the value by name. Variables enable the flexibility to represent different situations, process different sets of data, and produce varying outputs.</p> <p><u>6–8. Algorithms and Programming. Control</u> Programmers select and combine control structures, such as loops, event handlers, and conditionals, to create more complex program behavior.</p> <p><u>6–8. Algorithms and Programming. Modularity</u> Programs use procedures to organize code, hide implementation details, and make code easier to reuse. Procedures can be repurposed in new programs. Defining parameters for procedures can generalize behavior and increase reusability.</p> <p><u>6–8. Algorithms and Programming. Program Development</u> People design meaningful solutions for others by defining a problem’s criteria and constraints, carefully considering the diverse needs and wants of the community, and testing whether criteria and constraints were met.</p>	<p><u>Level 1B</u></p> <p><u>Algorithms and Programming</u></p> <p>1B-AP-08 Compare and refine multiple algorithms for the same task and determine which is the most appropriate.</p> <p>1B-AP-09 Create programs that use variables to store and modify data.</p> <p>1B-AP-10 Create programs that include sequences, events, loops, and conditionals.</p> <p>1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.</p> <p>1B-AP-12 Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.</p> <p>1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences</p> <p>1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs.</p>

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
			<p>1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.</p> <p>1B-AP-16 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development.</p> <p>1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.</p> <p><u>Level 2</u></p> <p>2-AP-14 Create procedures with parameters to organize code and make it easier to reuse.</p> <p>2-AP-16 Incorporate existing code, media, and libraries into original programs, and give attribution.</p>
HTML: Levels 5-6			
Introduction to HTML: Tags and text	To learn how to get started with HTML by adding paragraphs of text to a page.		
Links			
Introduction to HTML: Adding images	To learn how to add images to a web page using HTML. To understand new vocabulary associated with using HTML, including: images, jpgs, text, headings and paragraphs.		
Links			
Introduction to HTML: Minibeasts page	To learn how to create a web page using headings, paragraphs and images.		
Links			
Introduction to HTML: Space page	To learn how to apply knowledge of HTML to create a web page using headings, paragraphs and images.		
Links		are used to determine the best path to send and receive data.	
Introduction to HTML: Shapes page	To learn how to apply knowledge of HTML to create a web page using headings, paragraphs and images.		
Links			
Introduction to HTML: Food glorious food	To create a simple web page about food using headings, paragraphs and images.		

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
Links			
HTML formatting and CSS: Changing colour	To learn how to change the colour of text using the colour property.		
Links			
HTML formatting and CSS: Fonts and Sizes	To learn how to change the size and font of text using the font-size and font-family properties.		
Links			
HTML formatting and CSS: Background colours and patterns	To learn how to change the 'background', 'margin' and 'padding' properties of different parts of a web page.		
Links			
HTML formatting and CSS: Dinosaurs Online	To learn how to apply knowledge of HTML to make a web page using text, headings, images and styling.		
Links			
HTML formatting and CSS: Space page	To learn how to apply knowledge of HTML to make a web page using text, headings, images and styling.		
Links			
HTML formatting and CSS: HTML standards	To learn more about how web developers structure their pages and HTML standards.		
Links			
HTML Links: Introducing Links	To learn how to add links to websites and pages.		
Links			
HTML Links: Links from pictures	To learn how to make a link using an image.		
Links			
HTML Links: Links within pages	To learn how to make a page with anchor tags and section IDs to navigate within the page.		
Links			

DE Coding Lesson	CAS Pathways	K-12 Computer Science Framework	CSTA K-12 Computer Science Standards
HTML Links: Making a meal of it	To learn how to use div tags within a web page.		
Links			
HTML Links: Sports and PE	Make a web page combining divs, images and anchor links.		
Links			
HTML Links: MyShop – your own online store	To create a web page that combines the use of div tags, styles and anchor links.		
Links			