Skill Struck ®

Standards Alignment + Correlation

Table of Contents

JavaScript 2

Python 1

Python 2

Skill Struck ®

A New Approach to Computer Science Education

Computer science education (CSE) is in an interesting place at the moment, with most schools working to provide CSE for their students. However, many struggle to implement CSE at every grade level due to:

1. a lack of CSE standards across all schools; and
2. CSE curriculum and software that fails students with diverse backgrounds, learning styles, learning needs, and previous experience with CSE.

The Skill Struck Approach

Each grade is built on the assumption that students in that track are starting with zero programming background. The second level lessons (HTML & CSS 2, Python 2, and JavaScript 2) all assume students have taken the first course.

Engage

When using Skill Struck's lesson plans, we recommend that teachers use the lesson plan that correlates to their current grade. That being said, any lesson plan can be used for any grade, depending on student ability level. Elementary grades utilize sprites to engage and inspire students.

Students in higher elementary grades are encouraged to start learning the basics of HTML 1. The Python 1, JavaScript 1 and HTML & CSS 1 courses are all designed for middle school students. No previous programming experience is necessary. High school classes are also welcome to use these courses, or move students on to the second level courses (HTML & CSS 2, Python 2, and JavaScript 2).

Reinforce

As students move through the courses, the repetition of programming principles will reinforce concepts, programming standards, and computational thinking. Repetition will help them produce higher quality projects that align with increasing industry standards.

Students who start their CSE later than others will make programs using the same concepts as their peers, but may not work on as many extra challenges as more experienced students.

They will still learn concepts required by State Standards. Most importantly, students in the same grade, regardless of their ability, will learn the same topics, at the same time.

Accommodating advanced students

Some students learn faster than others, or bring a larger CSE background into the classroom. We understand the importance of encouraging their accelerated learning alongside their day-to-day lessons.

If students feel confident in their abilities, they can supplement their in-class lessons with additional challenges and other supplemental curriculum provided by Skill Struck. Doing so will keep advanced learners engaged while allowing new students to work at a comfortable pace.

Supplemental materials include challenges, projects, and advanced HTML/ CSS units (CSS 1, HTML & CSS 2).

Grades 10 - 12

Grades 10 through 12 are encouraged to use whatever Skill Struck lessons fit the needs of the school and the student. Some schools will need to choose first level lessons such as Python 1, JavaScript 1, or HTML & CSS 1. Other schools will choose the second or third level lessons at this point.

In the written lesson plans for grades 11 and 12, it is assumed students will have some JavaScript 1 experience. If not, we've allotted time during the Python 1 and Python 2 units for them to catch up using the JavaScript 1 unit.

Kindergarten

Course Overview and Goals

The Kindergarten course teaches the basics of what a computer is and foundational computer science concepts. Students will practice algorithmic thinking and critical problem solving through applicable activities. Through both plugged and unplugged activities and challenges, students will exercise logical thinking processes.

Who is it for?

The Kindergarten course is designed for students in Kindergarten. No previous programming or computer experience is required.

01 What is a Computer?

Topics:

- Computer Parts
- Computer Software
- Computer Identification

Standards:

1A-CS-03: Describe basic hardware and software problems using accurate terminology.

1A-CS-02: Use appropriate terminology in identifying and describing the function of common physical components of computing systems (hardware).

1A-CS-01: Select and operate appropriate software to perform a variety of tasks, and recognize that users have different needs and preferences for the technology they use.

02 Computer Devices

Topics:

- Computer Device Identification
- Computers and Daily Life
- Creative Attribution

Standards:

1A-IC-16: Compare how people live and work before and after the implementation or adoption of new computing technology.

1A-AP-13: Give attribution when using the ideas and creations of others while developing programs.

1A-CS-01: Select and operate appropriate software to perform a variety of tasks, and recognize that users have different needs and preferences for the technology they use.

03 Digital Citizenship

Topics:

- Being respectful
- Being responsible

Standards:

1A-IC-17: Work respectfully and responsibly with others online.

04 Computer Navigation

Topics:

- File Creation
- Sorting
- Filing
- Password Safety

Standards:

1A-IC-18: Keep login information private, and log off of devices appropriately.

1A-DA-05: Store, copy, search, retrieve, modify, and delete information using a computing device and define the information stored as data.

1A-NI-04: Explain what passwords are and why we use them, and use strong passwords to protect devices and information from unauthorized access.

1A-IC-17: Work respectfully and responsibly with others online.

05 Algorithms

Topics:

- Algorithm Creation
- Algorithm Debugging

- Algorithm Analysis

Standards:

1A-AP-14: Debug (identify and fix) errors in an algorithm or program that includes sequences and simple loops.

1A-AP-12: Develop plans that describe a program's sequence of events, goals, and expected outcomes.

1A-AP-08: Model daily processes by creating and following algorithms (sets of step-by-step instructions) to complete tasks.

06 Data

Topics:

- Store and Manipulate Data
- Generate Predictions from Data
- Collection and Presentation of Data

Standards:

1A-AP-09: Model the way programs store and manipulate data by using numbers or other symbols to represent information.

1A-DA-07: Identify and describe patterns in data visualizations, such as charts or graphs, to make predictions.

1A-DA-06: Collect and present the same data in various visual formats.

07 Weather Data

Topics:

- Making predictions
- Collecting data
- Identifying patterns
- Designing a shade structure

Standards:

1A-AP-09: Model the way programs store and manipulate data by using numbers or other symbols to represent information.

1B-DA-06: Organize and present collected data visually to highlight relationships and support a claim.

1B-DA-07: Use data to highlight or propose cause-and-effect relationships, predict outcomes, or communicate an idea.

08 Algorithmic Thinking

Topics:

- Sequencing Planning
- Debugging
- Step By Step Algorithms

Standards:

1A-AP-15: Using correct terminology, describe steps taken and choices made during the iterative process of program development.

1A-AP-14: Debug (identify and fix) errors in an algorithm or program that includes sequences and simple loops.

1A-AP-12: Develop plans that describe a program's sequence of events, goals, and expected outcomes.

1A-AP-10: Develop programs with sequences and simple loops, to express ideas or address a problem.

A Algorithm Practice (Optional)

Topics:

- Generating Algorithms
- Sequencing

Standards:

1A-AP-08: Model daily processes by creating and following algorithms (sets of step-by-step instructions) to complete tasks.

1A-DA-07: Identify and describe patterns in data visualizations, such as charts or graphs, to make predictions.

B Data Practice

Topics:

- Pattern Identification

- Data Representation

Standards:

1A-IC-17: Work respectfully and responsibly with others online.

1A-IC-16: Compare how people live and work before and after the implementation or adoption of new computing technology.

C Communication

Topics:

- Computer Network
- Computer Communication

Standards

1A-IC-17: Work respectfully and responsibly with others online.

1A-IC-16: Compare how people live and work before and after the implementation or adoption of new computing technology.

D Computational Thinking

Topics:

- Sequencing Creation
- Critical Thinking
- Breaking Down Big Ideas

Standards:

Skill Struck ®

1A-AP-14: Debug (identify and fix) errors in an algorithm or program that includes sequences and simple loops.

1A-AP-11: Decompose (break down) the steps needed to solve a problem into a precise sequence of instructions.

1A-AP-08: Model daily processes by creating and following algorithms (sets of step-by-step instructions) to complete tasks.

1st Grade

Course Overview and Goals

The 1st Grade course teaches the basics of what a computer is and foundational computer science concepts. Students will practice algorithmic thinking and critical problem solving through applicable activities and programming on the Scratch website. Through both plugged and unplugged activities and challenges, students will exercise logical thinking processes.

Who is it for?

The 1st Grade course is designed for students in the 1st grade. No previous programming or computer experience is required.

01 What is Respect?

Topics:

- Use Respect Online and In Person
- Computer Etiquette
- Computer Responsibility

Standards:

1A-IC-17: Work respectfully and responsibly with others online.

02 Data

Topics:

- Data Representation
- Data Visualization
- Predictions Based on Data

Standards:

1A-AP-09: Model the way programs store and manipulate data by using numbers or other symbols to represent information.

1A-DA-07: Identify and describe patterns in data visualizations, such as charts or graphs, to make predictions.

1A-DA-06: Collect and present the same data in various visual formats.

03 Algorithms

Topics:

- Algorithm Exploration
- Sequencing
- Debugging of Processes

Standards:

1A-AP-11: Decompose (break down) the steps needed to solve a problem into a precise sequence of instructions.

1A-AP-08: Model daily processes by creating and following algorithms (sets of step-by-step instructions) to complete tasks.

1A-AP-15: Using correct terminology, describe steps taken and choices made during the iterative process of program development.

04 Computer Parts

Topics:

- Hardware and Software
- Exploration of Computer Parts and Pieces
- Basic Troubleshooting

Standards

1A-AP-11: Decompose (break down) the steps needed to solve a problem into a precise sequence of instructions.

1A-AP-08: Model daily processes by creating and following algorithms (sets of step-by-step instructions) to complete tasks.

1A-AP-15: Using correct terminology, describe steps taken and choices made during the iterative process of program development.

05 Computer Organization Basics

Topics:

- Password Basics
- Computer Software
- File Navigation
- Attribution of Creative Works

Standards:

1A-IC-18: Keep login information private, and log off of devices appropriately.

1A-DA-05: Store, copy, search, retrieve, modify, and delete information using a computing device and define the information stored as data.

1A-NI-04: Explain what passwords are and why we use them, and use strong passwords to protect devices and information from unauthorized access.

1A-CS-01: Select and operate appropriate software to perform a variety of tasks, and recognize that users have different needs and preferences for the technology they use.

06 Introduction to Scratch

Topics:

- Sequencing
- Algorithm Building
- Debugging
- Break Down Problems Standards

Standards:

1A-AP-14: Debug (identify and fix) errors in an algorithm or program that includes sequences and simple loops.

1A-AP-12: Develop plans that describe a program's sequence of events, goals, and expected outcomes.

1A-AP-11: Decompose (break down) the steps needed to solve a problem into a precise sequence of instructions.

1A-AP-10: Develop programs with sequences and simple loops, to express ideas or address a problem.

07 Scratch Exercises

Topics:

- Sequencing
- Algorithm Building
- Loops
- Debugging Standards

Standards:

1A-AP-14: Debug (identify and fix) errors in an algorithm or program that includes sequences and simple loops.

1A-AP-12: Develop plans that describe a program's sequence of events, goals, and expected outcomes.

1A-AP-11: Decompose (break down) the steps needed to solve a problem into a precise sequence of instructions.

1A-AP-10: Develop programs with sequences and simple loops, to express ideas or address a problem.

08 Computer Device Exploration

Topics:

- Computer Device Selection
- Troubleshooting of Various Devices
- Hardware & Software Standards

Standards

1A-IC-16: Compare how people live and work before and after the implementation or adoption of new computing technology.

1A-CS-03: Describe basic hardware and software problems using accurate terminology.

1A-CS-02: Use appropriate terminology in identifying and describing the function of common physical components of computing systems (hardware).

A Bar Chart

Topics:

- Data Representation
- Data Tracking
- Building Predictions Based on Data Standards

Standards:

1A-DA-07: Identify and describe patterns in data visualizations, such as charts or graphs, to make predictions.

1A-DA-06: Collect and present the same data in various visual formats.

B Algorithm Practice

Topics:

- Building Algorithms
- Debugging
- Breaking Down Problems Standards

Standards:

1A-AP-11: Decompose (break down) the steps needed to solve a problem into a precise sequence of instructions.

1A-AP-10: Develop programs with sequences and simple loops, to express ideas or address a problem.

1A-AP-08: Model daily processes by creating and following algorithms (sets of step-by-step instructions) to complete tasks.

C Crayon Chart

Topics:

- Pattern Predictions
- Data Representations Standards

Standards:

1A-DA-07: Identify and describe patterns in data visualizations, such as charts or graphs, to make predictions.

1A-DA-06: Collect and present the same data in various visual formats.

Skill Struck ®

D Further Algorithm Practice

Topics:

- Sequencing
- Algorithms
- Debugging

Standards:

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-11: Decompose (break down) the steps needed to solve a problem into a precise sequence of instructions.

2nd Grade

Course Overview and Goals

The 2nd Grade course teaches the basics of what a computer is and foundational computer science concepts. Students will practice algorithmic thinking and critical problem solving through applicable activities and programming on the Scratch website. Through both plugged and unplugged activities and challenges, students will exercise logical thinking processes.

Who is it for?

The 2nd Grade course is designed for students in the 2nd grade. No previous programming or computer experience is required.

01 Technology in Our Lives

Topics:

- Society and Technology
- Impact of Technology

Standards:

1A-IC-16: Compare how people live and work before and after the implementation or adoption of new computing technology.

02 Crediting Others

Topics:

- Giving Credit Scenario
- Art Project Practice

Standards:

1A-IC-17: Work respectfully and responsibly with others online.

1A-AP-13: Give attribution when using the ideas and creations of others while developing programs.

03 Passwords

Topics:

- Password Safety
- PLUM Passwords
- Password Creation

Standards:

1A-IC-18: Keep login information private, and log off of devices appropriately.

1A-NI-04: Explain what passwords are and why we use them, and use strong passwords to protect devices and information from unauthorized access.

04 File Organization

Topics:

- File Creation
- File Location

- Computer Navigation

Standards:

1A-DA-05: Store, copy, search, retrieve, modify, and delete information using a computing device and define the information stored as data.

05 Troubleshooting

Topics:

- Hardware vs Software
- Computer Parts
- Basic Troubleshooting

Standards:

1A-CS-01: Select and operate appropriate software to perform a variety of tasks, and recognize that users have different needs and preferences for the technology they use.

1A-CS-03: Describe basic hardware and software problems using accurate terminology.

1A-CS-02: Use appropriate terminology in identifying and describing the function of common physical components of computing systems (hardware).

06 Data

Topics:

- Data Collection
- Data Representation
- Data Analysis

Standards:

1A-AP-09: Model the way programs store and manipulate data by using numbers or other symbols to represent information.

1A-DA-07: Identify and describe patterns in data visualizations, such as charts or graphs, to make predictions.

1A-DA-06: Collect and present the same data in various visual formats.

07 Getting Started with Scratch

Topics:

- Login Practice
- Exploration of the Scratch Tool

Standards:

1A-IC-18: Keep login information private, and log off of devices appropriately.

1A-NI-04: Explain what passwords are and why we use them, and use strong passwords to protect devices and information from unauthorized access.

1A-AP-15: Using correct terminology, describe steps taken and choices made during the iterative process of program development.

1A-AP-14: Debug (identify and fix) errors in an algorithm or program that includes sequences and simple loops.

1A-AP-12: Develop plans that describe a program’s sequence of events, goals, and expected outcomes.

1A-AP-11: Decompose (break down) the steps needed to solve a problem into a precise sequence of instructions.

1A-AP-10: Develop programs with sequences and simple loops, to express ideas or address a problem.

08 Algorithms

Topics:

- Building Algorithms
- Debugging Programs
- Breaking Down Problems into Steps

Standards:

1A-AP-15: Using correct terminology, describe steps taken and choices made during the iterative process of program development.

1A-AP-14: Debug (identify and fix) errors in an algorithm or program that includes sequences and simple loops.

1A-AP-12: Develop plans that describe a program’s sequence of events, goals, and expected outcomes.

1A-AP-11: Decompose (break down) the steps needed to solve a problem into a precise sequence of instructions.

1A-AP-10: Develop programs with sequences and simple loops, to express ideas or address a problem.

1A-AP-08: Model daily processes by creating and following algorithms (sets of step-by-step instructions) to complete tasks.

09 Program Planning

Topics:

- Building Algorithms
- Planning out Programs
- Debugging

Standards:

1A-AP-15: Using correct terminology, describe steps taken and choices made during the iterative process of program development.

1A-AP-14: Debug (identify and fix) errors in an algorithm or program that includes sequences and simple loops.

1A-AP-12: Develop plans that describe a program's sequence of events, goals, and expected outcomes.

1A-AP-11: Decompose (break down) the steps needed to solve a problem into a precise sequence of instructions.

1A-AP-10: Develop programs with sequences and simple loops, to express ideas or address a problem.

10 Team Programming

Topics:

- Building Algorithms
- Planning out Programs
- Debugging
- Collaboration with Teammates

Standards:

1A-IC-17: Work respectfully and responsibly with others online.

1A-AP-14: Debug (identify and fix) errors in an algorithm or program that includes sequences and simple loops.

1A-AP-13: Give attribution when using the ideas and creations of others while developing programs.

1A-AP-12: Develop plans that describe a program's sequence of events, goals, and expected outcomes.

1A-AP-11: Decompose (break down) the steps needed to solve a problem into a precise sequence of instructions.

1A-AP-10: Develop programs with sequences and simple loops, to express ideas or address a problem.

11 Team Programming

Topics:

- Building Algorithms
- Planning out Programs
- Debugging
- Collaboration with Teammates

Standards:

1A-IC-17: Work respectfully and responsibly with others online.

1A-AP-14: Debug (identify and fix) errors in an algorithm or program that includes sequences and simple loops.

1A-AP-13: Give attribution when using the ideas and creations of others while developing programs.

1A-AP-12: Develop plans that describe a program's sequence of events, goals, and expected outcomes.

1A-AP-11: Decompose (break down) the steps needed to solve a problem into a precise sequence of instructions.

1A-AP-10: Develop programs with sequences and simple loops, to express ideas or address a problem.

3rd Grade

Course Overview and Goals

The 3rd Grade course teaches the basics of what a computer is and foundational computer science concepts. Students will practice algorithmic thinking and critical problem solving through applicable activities. Students will be designing algorithms as they plan, debug, and implement programs. Through both plugged and unplugged activities and challenges, students will exercise logical thinking processes. **Cross curriculum is available.*

Who is it for?

The 3rd Grade course is designed for students in the 3rd grade. No previous programming or computer experience is required.

01 Computers and Culture

Topics:

- Exploring Computers and Your World
- Computer Identification

Standards:

1B-IC-20: Seek diverse perspectives for the purpose of improving computational artifacts.

1B-IC-18: Discuss computing technologies that have changed the world, and express how those technologies influence, and are influenced by, cultural practices.

02 Sequencing and Algorithms

Topics:

- Placing Steps in Order

Platform activities:

- Sequencing and Algorithms

Standards:

1B-AP-17: Describe choices made during program development using code comments, presentations, and demonstrations.

1B-AP-15: Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.

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-11: Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.

1B-AP-10: Create programs that include sequences, events, loops, and conditionals.

1B-AP-08: Compare and refine multiple algorithms for the same task and determine which is the most appropriate.

03 Computers as a System

Topics:

- Networks

Platform activities:

- Events

Standards:

1B-CS-01: Describe how internal and external parts of computing devices function to form a system.

1B-CS-02: Model how computer hardware and software work together as a system to accomplish tasks.

1B-NI-04: Model how information is broken down into smaller pieces, transmitted as packets through multiple devices over networks and the Internet, and reassembled at the destination.

1B-AP-08: Compare and refine multiple algorithms for the same task and determine which is the most appropriate.

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-15: Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.

04 Accessibility

Topics:

- Digital Accessibility

Platform activities:

- Variables

Standards:

1B-CS-01: Describe how internal and external parts of computing devices function to form a system.

1B-NI-04: Model how information is broken down into smaller pieces, transmitted as packets through multiple devices over networks and the Internet, and reassembled at the destination.

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-11: Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.

1B-AP-13: Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences.

1B-AP-15: Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.

1B-IC-18: Discuss computing technologies that have changed the world, and express how those technologies influence, and are influenced by, cultural practices.

1B-IC-19: Brainstorm ways to improve the accessibility and usability of technology products for the diverse needs and wants of users.

1B-IC-20: Seek diverse perspectives for the purpose of improving computational artifacts.

05 Computer Safety

Topics:

- Data Protection

Platform activities:

- Online Safety

Standards:

1B-IC-21: Use public domain or creative commons media, and refrain from copying or using material created by others without permission.

1B-IC-18: Discuss computing technologies that have changed the world, and express how those technologies influence, and are influenced by, cultural practices.

1B-AP-14: Observe intellectual property rights and give appropriate attribution when creating or remixing programs.

1B-NI-05: Discuss real-world cybersecurity problems and how personal information can be protected.

06 Troubleshooting

Topics:

- Hardware vs Software
- Computer Troubleshooting

Platform activities:

- Computer Parts

Standards:

1B-AP-11: Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.

1B-CS-03: Determine potential solutions to solve simple hardware and software problems using common troubleshooting strategies.

1B-CS-02: Model how computer hardware and software work together as a system to accomplish tasks.

1B-CS-01: Describe how internal and external parts of computing devices function to form a system.

07 Conditionals

Topics:

- Class Rules

Platform activities:

- Conditionals

Standards:

1B-AP-17: Describe choices made during program development using code comments, presentations, and demonstrations.

1B-AP-15: Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.

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-10: Create programs that include sequences, events, loops, and conditionals.

1B-AP-09: Create programs that use variables to store and modify data.

1B-AP-08: Compare and refine multiple algorithms for the same task and determine which is the most appropriate.

08 Loops

Topics:

- Charade Loops

Platform activities:

- Loops

Standards:

1B-AP-15: Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.

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-11: Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.

1B-AP-10: Create programs that include sequences, events, loops, and conditionals.

1B-AP-09: Create programs that use variables to store and modify data.

1B-AP-08: Compare and refine multiple algorithms for the same task and determine which is the most appropriate.

09 Property Rights

Topics:

- Property Rights
- Loop Line

Platform activities:

- Loops

Standards:

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-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-IC-21: Use public domain or creative commons media, and refrain from copying or using material created by others without permission.

10 Data and Graphs

Topics:

- Weather Patterns
- Data Analysis

Standards:

1B-DA-07: Use data to highlight or propose cause-and-effect relationships, predict outcomes, or communicate an idea.

1B-DA-06: Organize and present collected data visually to highlight relationships and support a claim.

11 Partner Programming

Topics:

- Collaboration
- Feedback

Platform activities:

- Choice

Standards:

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-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.

12 Partner Programming

Topics:

- Collaboration
- Feedback

Platform activities:

- Choice

Standards:

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-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.

A Cross Curriculum: Exploring Science Through Coding

01 HTML and Life Cycles

Topics:

- What HTML is
- HTML tags with block coding
- The life cycle of plants and animals

Standards:

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-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-15: Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.

NGSS 3-LS1-1: Develop models to describe that organisms have unique and diverse life cycles but all have in common birth, growth, reproduction, and death.

NGSS 3-5-ETS1-2: Generate and compare multiple possible solutions to a problem based on how well each is likely to meet the criteria and constraints of the problem.

02 Paragraphs and Climates

Topics:

- Paragraph tags with block coding
- Weather vs climate

Standards:

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-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-15: Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.

NGSS 3-ESS2-2: Obtain and combine information to describe climates in different regions of the world.

NGSS 3-5-ETS1-2: Generate and compare multiple possible solutions to a problem based on how well each is likely to meet the criteria and constraints of the problem.

03 Headers and Heredity

Topics:

- Header tags with block coding
- Header hierarchy
- Heredity
- Animals' survival traits
- The environment's impact on animals

Standards:

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-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-15: Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.

NGSS 3-LS3-1: Analyze and interpret data to provide evidence that plants and animals have traits inherited from parents and that variation of these traits exists in a group of similar organisms.

NGSS 3-LS3-2: Use evidence to support the explanation that traits can be influenced by the environment.

NGSS 3-5-ETS1-2: Generate and compare multiple possible solutions to a problem based on how well each is likely to meet the criteria and constraints of the problem.

04 Images and Animal Survival

Topics :

- Finding images and copyright
- Linking images and image tags
- Change the image size
- Species in different ecosystems
- Animals survive in groups

Standards:

1B-AP-09: Create programs that use variables to store and modify data.

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-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.

NGSS 3-LS2-1: Construct an argument that some animals form groups that help members survive.

NGSS 3-5-ETS1-2: Generate and compare multiple possible solutions to a problem based on how well each is likely to meet the criteria and constraints of the problem.

B Cross Curriculum: Exploring Math Through Coding

01 HTML Basics and Perimeter

Topics:

- What HTML is
- HTML tags with block coding
- Perimeter

Standards:

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-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-15: Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.

CCSS.MATH.CONTENT.3.MD.D.8: Solve real world and mathematical problems involving perimeters of polygons.

CCSS.MATH.PRACTICE.MP1: Make sense of problems and persevere in solving them.

CCSS.MATH.PRACTICE.MP2: Reason abstractly and quantitatively.

CCSS.MATH.PRACTICE.MP4: Model with mathematics.

CCSS.MATH.PRACTICE.MP5: Use appropriate tools strategically.

CCSS.MATH.PRACTICE.MP6: Attend to precision.

CCSS.MATH.PRACTICE.MP7: Look for and make use of structure.

02 Sequencing in Programming and Math

Topics:

- What sequencing is
- Sequence order
- Sequencing in programming
- Using sequencing to solve word problems

Standards:

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.

1B-AP-15: Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.

CCSS.MATH.CONTENT.3.OA.A.3: Use multiplication and division within 100 to solve word problems in situations involving equal groups, arrays, and measurement quantities.

CCSS.MATH.PRACTICE.MP1: Make sense of problems and persevere in solving them.

CCSS.MATH.PRACTICE.MP2: Reason abstractly and quantitatively.

CCSS.MATH.PRACTICE.MP4: Model with mathematics.

CCSS.MATH.PRACTICE.MP6: Attend to precision.

CCSS.MATH.PRACTICE.MP7: Look for and make use of structure.

03 Variables in Programming and Math

Topics:

- Variables in programming

- Multiplying with variables
- Dividing with variables

Standards:

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-15: Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.

CCSS.MATH.CONTENT.3.OA.A.4: Determine the unknown whole number in a multiplication or division equation relating three whole numbers.

CCSS.MATH.PRACTICE.MP1: Make sense of problems and persevere in solving them.

CCSS.MATH.PRACTICE.MP2: Reason abstractly and quantitatively.

CCSS.MATH.PRACTICE.MP4: Model with mathematics.

CCSS.MATH.PRACTICE.MP6: Attend to precision.

CCSS.MATH.PRACTICE.MP7: Look for and make use of structure.

04 Conditionals in Programming and Math

Topics:

- Conditionals in programming
- Conditionals in math
- Rounding
- Place value

Standards:

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-15: Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.

CCSS.MATH.CONTENT.3.NBT.A.1: Use place value understanding to round whole numbers to the nearest 10 or 100.

CCSS.MATH.PRACTICE.MP1: Make sense of problems and persevere in solving them.

CCSS.MATH.PRACTICE.MP2: Reason abstractly and quantitatively.

CCSS.MATH.PRACTICE.MP4: Model with mathematics.

CCSS.MATH.PRACTICE.MP6: Attend to precision.

CCSS.MATH.PRACTICE.MP7: Look for and make use of structure.

C Cross Curriculum: Exploring English Language Arts Through Coding

01 HTML Basics and Tiles

Topics:

- What HTML is
- HTML tags with block coding
- Writing tiles

Standards:

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-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-15: Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.

CCSS.ELA-LITERACY.L.3.2.A: Capitalize appropriate words in titles.

02 Headers and Vocabulary

Topics:

- Header tags
- Header hierarchy
- Prefixes
- Suffixes

Standards:

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-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-15: Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.

CCSS.ELA-LITERACY.L.3.4.B: Determine the meaning of the new word formed when a known affix is added to a known word.

Skill Struck ®

03 Paragraphs

Topics:

- Paragraph tags
- Writing paragraphs

Standards:

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-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-15: Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.

CCSS.ELA-LITERACY.L.3.4.A: Use sentence-level context as a clue to the meaning of a word or phrase.

CCSS.ELA-LITERACY.L.3.5.B: Identify real-life connections between words and their use

Skill Struck ®

4th Grade

Course Overview and Goals

The 4th Grade course teaches the basics of what a computer is and foundational computer science concepts. Students will practice algorithmic thinking and critical problem solving through applicable activities. Students will be designing algorithms as they plan, debug, and implement programs. Through both plugged and unplugged activities and challenges, students will exercise logical thinking processes.

Who is it for?

The 4th Grade course is designed for students in the 4th grade. No previous programming or computer experience is required.

01 Computers and Culture

Topics:

- Amazon Rainforest Culture
- Culture Exploration

Standards:

1B-IC-19: Brainstorm ways to improve the accessibility and usability of technology products for the diverse needs and wants of users.

02 Sequencing and Algorithms

Topics:

- Corridor

Platform activities:

- Sequencing and Algorithms

Standards:

1B-AP-15: Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.

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-11: Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.

1B-AP-10: Create programs that include sequences, events, loops, and conditionals.

1B-AP-08: Compare and refine multiple algorithms for the same task and determine which is the most appropriate.

03 Computers as a System

Topics:

- How the Internet Works

Platform activities:

- Events in Programming

Standards:

1B-CS-02: Model how computer hardware and software work together as a system to accomplish tasks.

1B-NI-04: Model how information is broken down into smaller pieces, transmitted as packets through multiple devices over networks and the Internet, and reassembled at the destination.

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.

1B-AP-15: Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.

04 Digital Citizenship

Topics:

- Computer Safety
- Digital Citizenship

Platform activities:

- Variables

Standards:

1B-NI-05: Discuss real-world cybersecurity problems and how personal information can be protected.

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-15: Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.

1B-IC-18: Discuss computing technologies that have changed the world, and express how those technologies influence, and are influenced by, cultural practices.

05 Computer Parts

Topics:

- Computer Pieces

Platform activities:

- Conditionals in Programming

Standards:

1B-CS-01: Describe how internal and external parts of computing devices function to form a system.

1B-CS-02: Model how computer hardware and software work together as a system to accomplish tasks.

1B-CS-03: Determine potential solutions to solve simple hardware and software problems using common troubleshooting strategies.

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-15: Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.

06 Accessibility

Topics:

- Digital Accessibility

Platform activities:

- Loops in Programming

Standards:

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-15: Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.

1B-IC-18: Discuss computing technologies that have changed the world, and express how those technologies influence, and are influenced by, cultural practices.

1B-IC-19: Brainstorm ways to improve the accessibility and usability of technology products for the diverse needs and wants of users.

1B-IC-20: Seek diverse perspectives for the purpose of improving computational artifacts.

07 Intro to HTML

Platform activities:

- Header Tags
- Paragraph Tags

Standards:

1B-AP-15: Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.

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-11: Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.

1B-AP-09: Create programs that use variables to store and modify data.

1B-AP-08: Compare and refine multiple algorithms for the same task and determine which is the most appropriate.

08 Color & Troubleshooting

Topics:

- Digital Troubleshooting

Platform activities:

- Text Color

Standards:

2-AP-15: Seek and incorporate feedback from team members and users to refine a solution that meets user needs.

2-AP-12: Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

2-DA-09: Refine computational models based on the data they have generated.

2-CS-03: Systematically identify and fix problems with computing devices and their components.

09 Images and Property Rights

Topics:

- Stealing Dice

Platform activities:

- Images

Standards:

1B-IC-21: Use public domain or creative commons media, and refrain from copying or using material created by others without permission.

1B-AP-17: Describe choices made during program development using code comments, presentations, and demonstrations.

1B-AP-14: Observe intellectual property rights and give appropriate attribution when creating or remixing programs.

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-11: Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.

1B-AP-09: Create programs that use variables to store and modify data.

10 Data & Graphs

Topics:

- Bar Graphs
- Data Analysis

Standards:

1B-DA-07: Use data to highlight or propose cause-and-effect relationships, predict outcomes, or communicate an idea.

1B-DA-06: Organize and present collected data visually to highlight relationships and support a claim.

11 Team Project

Topics:

- Collaboration
- Feedback

Standards:

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-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.

1B-IC-20: Seek diverse perspectives for the purpose of improving computational artifacts.

1B-IC-21: Use public domain or creative commons media, and refrain from copying or using material created by others without permission.

12 Team Project

Topics:

- Collaboration
- Feedback

Standards:

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-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.

1B-IC-20: Seek diverse perspectives for the purpose of improving computational artifacts.

1B-IC-21: Use public domain or creative commons media, and refrain from copying or using material created by others without permission.

A Cross Curriculum: Exploring Science Through Coding

01 HTML and the Structures of Living Organisms

Topics:

- What HTML is
- HTML tags
- External structures of living organisms
- Internal structures of living organisms

Standards:

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-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-15: Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.

NGSS 4-LS1-1: Construct an argument that plants and animals have internal and external structures that function to support survival, growth, behavior, and reproduction.

NGSS 3-5-ETS1-2: Generate and compare multiple possible solutions to a problem based on how well each is likely to meet the criteria and constraints of the problem.

02 Headers and Rock Formations

Topics:

- Header tags
- Header hierarchy
- Weathering and erosion
- Fossils

Standards:

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-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-15: Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.

NGSS 4-ESS1-1: Identify evidence from patterns in rock formations and fossils in rock layers to support an explanation for changes in a landscape over time.

NGSS 4-ESS2-1: Make observations and/or measurements to provide evidence of the effects of weathering or the rate of erosion by water, ice, wind, or vegetation.

NGSS 3-5-ETS1-2: Generate and compare multiple possible solutions to a problem based on how well each is likely to meet the criteria and constraints of the problem.

03 Text Color and Waves

Topics:

- Attributes
- Adding text color
- Amplitude and wavelength

- Earthquakes are examples of waves

Standards:

1B-AP-09: Create programs that use variables to store and modify data.

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-15: Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.

NGSS 4-PS4-1: Develop a model of waves to describe patterns in terms of amplitude and wavelength and that waves can cause objects to move.

NGSS 3-5-ETS1-2: Generate and compare multiple possible solutions to a problem based on how well each is likely to meet the criteria and constraints of the problem.

04 Paragraphs and Natural Resources

Topics:

- Paragraph tags
- How natural resources are used
- The impacts of natural resources
- Renewable resources

Standards:

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-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-15: Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.

NGSS 4-ESS3-1: Obtain and combine information to describe that energy and fuels are derived from natural resources and their uses affect the environment.

NGSS 3-5-ETS1-2: Generate and compare multiple possible solutions to a problem based on how well each is likely to meet the criteria and constraints of the problem.

05 Images and Reducing the Impact of Earth’s Processes

Topics:

- Finding images and copyright
- Linking images and image tags
- Change the image size
- Earthquakes
- Volcanoes
- Limiting a volcanoes impact

Standards:

1B-AP-09: Create programs that use variables to store and modify data.

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-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-IC-21: Use public domain or creative commons media, and refrain from copying or using material created by others without permission.

NGSS 4-ESS3-2: Generate and compare multiple solutions to reduce the impacts of natural Earth processes on humans.

NGSS 3-5-ETS1-2: Generate and compare multiple possible solutions to a problem based on how well each is likely to meet the criteria and constraints of the problem.

06 Science Fair Project

Topics:

- Application of learned skills

Standards:

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-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-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-IC-21: Use public domain or creative commons media, and refrain from copying or using material created by others without permission.

NGSS 4-LS1-1: Construct an argument that plants and animals have internal and external structures that function to support survival, growth, behavior, and reproduction.

NGSS 3-5-ETS1-2: Generate and compare multiple possible solutions to a problem based on how well each is likely to meet the criteria and constraints of the problem.

B Cross Curriculum: Exploring Math Through Coding

01 Sequencing in Programming and Math

Topics:

- Sequencing and order
- Sequences and algorithms in programming
- Completing patterns that follow a given rule

Standards:

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.

1B-AP-15: Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.

CCSS.MATH.CONTENT.4.OA.C.5: Generate a number or shape pattern that follows a given rule. Identify apparent features of the pattern that were not explicit in the rule itself.

CCSS.MATH.PRACTICE.MP1: Make sense of problems and persevere in solving them.

CCSS.MATH.PRACTICE.MP2: Reason abstractly and quantitatively.

CCSS.MATH.PRACTICE.MP4: Model with mathematics.

CCSS.MATH.PRACTICE.MP6: Attend to precision.

CCSS.MATH.PRACTICE.MP7: Look for and make use of structure.

02 Variables in Programming and Math

Topics:

- Variables in programming
- Multiplying with variables
- Dividing with variables
- Multi step word problems using the four operations

Standards:

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-15: Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.

CCSS.MATH.CONTENT.4.OA.A.2: Multiply or divide to solve word problems involving multiplicative comparison

CCSS.MATH.CONTENT.4.OA.A.3: Solve multistep word problems posed with whole numbers and having whole-number answers using the four operations, including problems in which remainders must be interpreted. Represent these problems using equations with a letter standing for the unknown quantity.

CCSS.MATH.PRACTICE.MP1: Make sense of problems and persevere in solving them.

CCSS.MATH.PRACTICE.MP2: Reason abstractly and quantitatively.

CCSS.MATH.PRACTICE.MP4: Model with mathematics.

CCSS.MATH.PRACTICE.MP6: Attend to precision.

CCSS.MATH.PRACTICE.MP7: Look for and make use of structure.

03 Conditionals in Programming and Math

Topics:

- Conditionals in programming
- Inequalities

Standards:

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-15: Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.

CCSS.MATH.CONTENT.4.NBT.A.2: Read and write multi-digit whole numbers using base-ten numerals, number names, and expanded form. Compare two multi-digit numbers based on meanings of the digits in each place, using >, =, and < symbols to record the results of comparisons.

CCSS.MATH.PRACTICE.MP1: Make sense of problems and persevere in solving them.

CCSS.MATH.PRACTICE.MP2: Reason abstractly and quantitatively.

CCSS.MATH.PRACTICE.MP4: Model with mathematics.

CCSS.MATH.PRACTICE.MP5: Use appropriate tools strategically.

CCSS.MATH.PRACTICE.MP6: Attend to precision.

CCSS.MATH.PRACTICE.MP7: Look for and make use of structure.

CCSS.MATH.PRACTICE.MP8: Look for and express regularity in repeated reasoning.

04 HTML and Word Problems

Topics:

- What HTML is
- HTML tags
- Using the four operations to solve word problems

Standards:

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-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-15: Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.

CCSS.MATH.CONTENT.4.MD.A.2: Use the four operations to solve word problems involving distances, intervals of time, liquid volumes, masses of objects, and money, including problems involving simple fractions or decimals, and problems that require expressing measurements given in a larger unit in terms of a smaller unit.

CCSS.MATH.PRACTICE.MP1: Make sense of problems and persevere in solving them.

CCSS.MATH.PRACTICE.MP2: Reason abstractly and quantitatively.

CCSS.MATH.PRACTICE.MP4: Model with mathematics.

CCSS.MATH.PRACTICE.MP5: Use appropriate tools strategically.

CCSS.MATH.PRACTICE.MP6: Attend to precision.

CCSS.MATH.PRACTICE.MP7: Look for and make use of structure.

05 Images and Geometry

Topics:

- Finding images and copyright
- Linking images and image tags
- Change the image size
- Identifying points, lines, line segments, rays, angles, and perpendicular and parallel lines in two-dimensional shapes

Standards:

1B-AP-09: Create programs that use variables to store and modify data.

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-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-IC-21: Use public domain or creative commons media, and refrain from copying or using material created by others without permission.

CCSS.MATH.CONTENT.4.G.A.1: Draw points, lines, line segments, rays, angles (right, acute, obtuse), and perpendicular and parallel lines. Identify these in two-dimensional figures.

CCSS.MATH.CONTENT.4.G.A.2: Classify two-dimensional figures based on the presence or absence of parallel or perpendicular lines, or the presence or absence of angles of a specified size. Recognize right triangles as a category, and identify right triangles.

CCSS.MATH.PRACTICE.MP1: Make sense of problems and persevere in solving them.

CCSS.MATH.PRACTICE.MP2: Reason abstractly and quantitatively.

CCSS.MATH.PRACTICE.MP4: Model with mathematics.

CCSS.MATH.PRACTICE.MP5: Use appropriate tools strategically.

CCSS.MATH.PRACTICE.MP6: Attend to precision.

CCSS.MATH.PRACTICE.MP7: Look for and make use of structure.

C Cross Curriculum: Exploring English Language Arts Through Coding

01 HTML Basics and Tiles

Topics:

- What HTML is
- HTML Tags
- Writing titles and correct capitalization

Standards:

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-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-15: Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.

CCSS.ELA-LITERACY.W.4.2.A: Introduce a topic clearly and group related information in paragraphs and sections; include formatting (e.g., headings), illustrations, and multimedia when useful to aiding comprehension.

CCSS.ELA-LITERACY.L.4.2.A: Use correct capitalization.

02 Headers and Subtitles

Topics:

- Header tags
- Header hierarchy
- Subtitles in writing

Standards:

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-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-15: Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.

CCSS.ELA-LITERACY.W.4.2.A: Introduce a topic clearly and group related information in paragraphs and sections; include formatting (e.g., headings), illustrations, and multimedia when useful to aiding comprehension.

CCSS.ELA-LITERACY.L.4.2.A: Use correct capitalization.

03 Paragraphs

Topics:

- Paragraph tags
- Writing hook sentences
- Writing introduction paragraphs
- Writing body paragraphs
- Writing conclusion paragraphs

Standards:

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-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-15: Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.

CCSS.ELA-LITERACY.W.4.1.A: Introduce a topic or text clearly, state an opinion, and create an organizational structure in which related ideas are grouped to support the writer's purpose.

CCSS.ELA-LITERACY.W.4.1.B: Provide reasons that are supported by facts and details.

CCSS.ELA-LITERACY.W.4.2.A: Introduce a topic clearly and group related information in paragraphs and sections; include formatting (e.g., headings), illustrations, and multimedia when useful to aiding comprehension.

CCSS.ELA-LITERACY.W.4.4: Produce clear and coherent writing in which the development and organization are appropriate to task, purpose, and audience.

CCSS.ELA-LITERACY.W.4.10: Write routinely over extended time frames and shorter time frames for a range of discipline-specific tasks, purposes, and audiences.

04 Opinion Writing

Topics:

- Application of learned skills
- Writing practice

Standards:

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-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-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-IC-21: Use public domain or creative commons media, and refrain from copying or using material created by others without permission.

CCSS.ELA-LITERACY.W.4.1: Write opinion pieces on topics or texts, supporting a point of view with reasons and information.

CCSS.ELA-LITERACY.W.4.4: Produce clear and coherent writing in which the development and organization are appropriate to task, purpose, and audience.

CCSS.ELA-LITERACY.W.4.5: With guidance and support from peers and adults, develop and strengthen writing as needed by planning, revising, and editing.

CCSS.ELA-LITERACY.W.4.6: With some guidance and support from adults, use technology, including the Internet, to produce and publish writing as well as to interact and collaborate with others; demonstrate sufficient command of keyboarding skills to type a minimum of one page in a single sitting.

CCSS.ELA-LITERACY.W.4.10: Write routinely over extended time frames and shorter time frames for a range of discipline-specific tasks, purposes, and audiences.

CCSS.ELA-LITERACY.L.4.1: Demonstrate command of the conventions of standard English grammar and usage when writing or speaking.

CCSS.ELA-LITERACY.L.4.2: Demonstrate command of the conventions of standard English capitalization, punctuation, and spelling when writing.

05 Narrative Writing

Topics:

- Application of learned skills
- Writing practice

Standards:

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-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-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-IC-21: Use public domain or creative commons media, and refrain from copying or using material created by others without permission.

CCSS.ELA-LITERACY.W.4.3: Write narratives to develop real or imagined experiences or events using effective technique, descriptive details, and clear event sequences.

CCSS.ELA-LITERACY.W.4.4: Produce clear and coherent writing in which the development and organization are appropriate to task, purpose, and audience.

CCSS.ELA-LITERACY.W.4.5: With guidance and support from peers and adults, develop and strengthen writing as needed by planning, revising, and editing.

CCSS.ELA-LITERACY.W.4.6: With some guidance and support from adults, use technology, including the Internet, to produce and publish writing as well as to interact and collaborate with others; demonstrate sufficient command of keyboarding skills to type a minimum of one page in a single sitting.

CCSS.ELA-LITERACY.W.4.8: Recall relevant information from experiences or gather relevant information from print and digital sources; take notes and categorize information, and provide a list of sources.

CCSS.ELA-LITERACY.W.4.10: Write routinely over extended time frames and shorter time frames for a range of discipline-specific tasks, purposes, and audiences.

CCSS.ELA-LITERACY.L.4.1: Demonstrate command of the conventions of standard English grammar and usage when writing or speaking.

CCSS.ELA-LITERACY.L.4.2: Demonstrate command of the conventions of standard English capitalization, punctuation, and spelling when writing.

06 Informational Writing

Topics:

- Application of learned skills
- Writing practice

Standards:

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-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-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-IC-21: Use public domain or creative commons media, and refrain from copying or using material created by others without permission.

CCSS.ELA-LITERACY.W.4.2: Write informative/explanatory texts to examine a topic and convey ideas and information clearly.

CCSS.ELA-LITERACY.W.4.4: Produce clear and coherent writing in which the development and organization are appropriate to task, purpose, and audience.

CCSS.ELA-LITERACY.W.4.5: With guidance and support from peers and adults, develop and strengthen writing as needed by planning, revising, and editing.

CCSS.ELA-LITERACY.W.4.6: With some guidance and support from adults, use technology, including the Internet, to produce and publish writing as well as to interact and collaborate with others; demonstrate sufficient command of keyboarding skills to type a minimum of one page in a single sitting.

CCSS.ELA-LITERACY.W.4.7: Conduct short research projects that build knowledge through investigation of different aspects of a topic.

CCSS.ELA-LITERACY.W.4.8: Recall relevant information from experiences or gather relevant information from print and digital sources; take notes and categorize information, and provide a list of sources.

CCSS.ELA-LITERACY.W.4.9: Draw evidence from literary or informational texts to support analysis, reflection, and research.

CCSS.ELA-LITERACY.W.4.10: Write routinely over extended time frames and shorter time frames for a range of discipline-specific tasks, purposes, and audiences.

CCSS.ELA-LITERACY.L.4.1: Demonstrate command of the conventions of standard English grammar and usage when writing or speaking.

CCSS.ELA-LITERACY.L.4.2: Demonstrate command of the conventions of standard English capitalization, punctuation, and spelling when writing.

CCSS.ELA-LITERACY.L.4.6: Acquire and use accurately grade-appropriate general academic and domain-specific words and phrases, including those that signal precise actions, emotions, or states of being and that are basic to a particular topic.

Skill Struck ®

5th Grade

01 Computers and Culture

Topics:

- Digital Culture
- Life without Technology

Platform activities:

- Sequencing and Algorithms

Standards:

1B-IC-18: Discuss computing technologies that have changed the world, and express how those technologies influence, and are influenced by, cultural practices.

1B-AP-15: Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.

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-11: Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.

1B-AP-10: Create programs that include sequences, events, loops, and conditionals.

1B-AP-08: Compare and refine multiple algorithms for the same task and determine which is the most appropriate.

02 Variables and Troubleshooting

Topics:

- Computer Parts
- Troubleshooting

Platform activities:

- Variables in Programming

Standards:

1B-CS-01: Describe how internal and external parts of computing devices function to form a system.

1B-CS-02: Model how computer hardware and software work together as a system to accomplish tasks.

1B-CS-03: Determine potential solutions to solve simple hardware and software problems using common troubleshooting strategies.

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-15: Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.

1B-AP-17: Describe choices made during program development using code comments, presentations, and demonstrations.

03 Improving Computational Artifacts

Topics:

- Analysis of Computational Artifacts

Platform activities:

- Loops in Programming

Standards:

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-15: Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.

1B-IC-20: Seek diverse perspectives for the purpose of improving computational artifacts.

04 Cybersecurity

Topics:

- Cybersecurity

Platform activities:

- Conditionals in Programming

Standards:

1B-NI-05: Discuss real-world cybersecurity problems and how personal information can be protected.

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-15: Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.

05 Data & Graphs

Topics:

- Data Analysis
- Pie Chart
- Data Representations

Standards:

1B-DA-07: Use data to highlight or propose cause-and-effect relationships, predict outcomes, or communicate an idea.

1B-DA-06: Organize and present collected data visually to highlight relationships and support a claim.

06 Intro to HTML

Platform activities

- Headers
- Text Colors

Standards:

1B-AP-15: Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.

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-11: Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.

1B-AP-09: Create programs that use variables to store and modify data.

07 The Internet

Topics:

- Internet Packets

Platform activities

- Fonts

Standards

1B-AP-15: Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.

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-11: Decompose (break down) problems into smaller, manageable

subproblems to facilitate the program development process.

1B-AP-09: Create programs that use variables to store and modify data.

1B-NI-04: Model how information is broken down into smaller pieces, transmitted as packets through multiple devices over networks and the Internet, and reassembled at the destination.

1B-CS-01: Describe how internal and external parts of computing devices function to form a system.

08 Accessibility

Topics:

- Digital Accessibility

Platform activities:

- Paragraphs
- Text Align

Standards:

1B-IC-20: Seek diverse perspectives for the purpose of improving computational artifacts.

1B-IC-19: Brainstorm ways to improve the accessibility and usability of technology products for the diverse needs and wants of users.

1B-AP-15: Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.

1B-AP-13: Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences.

1B-AP-11: Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.

1B-AP-09: Create programs that use variables to store and modify data.

09 Property Rights

Topics:

- Giving Credit

Platform activities:

- Images
- Videos

Standards:

1B-IC-21: Use public domain or creative commons media, and refrain from copying or using material created by others without permission.

1B-AP-15: Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.

1B-AP-14: Observe intellectual property rights and give appropriate attribution when creating or remixing programs.

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-11: Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.

1B-AP-09: Create programs that use variables to store and modify data.

10 Personal Project

Topics:

- Integration of Concepts Learned

Platform activities:

- Review of HTML Concepts
- Project Planning

Standards:

1B-AP-09: Create programs that use variables to store and modify data.

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-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-17: Describe choices made during program development using code comments, presentations, and demonstrations.

1B-IC-21: Use public domain or creative commons media, and refrain from copying or using material created by others without permission.

11 Team Project

Topics:

- Collaboration/Cooperation
- Feedback
- Integration of Concepts Learned

Standards:

1B-AP-09: Create programs that use variables to store and modify data.

1B-AP-11: Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.

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.

1B-IC-20: Seek diverse perspectives for the purpose of improving computational artifacts.

1B-IC-21: Use public domain or creative commons media, and refrain from copying or using material created by others without permission.

12 Team Project

Topics:

- Collaboration/Cooperation
- Feedback
- Integration of Concepts Learned

Standards:

1B-AP-09: Create programs that use variables to store and modify data.

1B-AP-11: Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.

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.

1B-IC-20: Seek diverse perspectives for the purpose of improving computational artifacts.

1B-IC-21: Use public domain or creative commons media, and refrain from copying or using material created by others without permission.

A Cross Curriculum: Exploring Science Through Coding

01 Headers and Earth’s Place in the Universe

Topics:

- Header tags
- Header hierarchy
- Apparent brightness of the sun and stars
- Patterns in shadows, seasons, day and night

Standards:

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-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-15: Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.

NGSS 5-ESS1-1: Support an argument that differences in the apparent brightness of the sun compared to other stars is due to their relative distances from the Earth.

NGSS 5-ESS1-2: Represent data in graphical displays to reveal patterns of daily changes in length and direction of shadows, day and night, and the seasonal appearance of some stars in the night sky.

NGSS 3-5-ETS1-2: Generate and compare multiple possible solutions to a problem based on how well each is likely to meet the criteria and constraints of the problem.

02 Text Colors and Organisms

Topics:

- Attributes
- Adding text color to tags

- Producers, consumers, and decomposers

Standards:

1B-AP-09: Create programs that use variables to store and modify data.

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-15: Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.

NGSS 5-LS2-1: Develop a model to describe the movement of matter among plants, animals, decomposers, and the environment.

NGSS 3-5-ETS1-2: Generate and compare multiple possible solutions to a problem based on how well each is likely to meet the criteria and constraints of the problem.

03 Paragraphs and Ecosystems

Topics:

- Paragraph tags
- How producers, consumers, and decomposers work together in the ecosystem

Standards:

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-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-15: Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.

NGSS 5-LS2-1: Develop a model to describe the movement of matter among plants, animals, decomposers, and the environment.

NGSS 3-5-ETS1-2: Generate and compare multiple possible solutions to a problem based on how well each is likely to meet the criteria and constraints of the problem.

04 Fonts and Gravity

Topics:

- Font families
- Font size
- Multiply styles in a single tag
- Gravity

Standards:

1B-AP-09: Create programs that use variables to store and modify data.

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-15: Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.

NGSS 5-PS2-1: Support an argument that the gravitational force exerted by Earth on objects is directed down.

NGSS 3-5-ETS1-2: Generate and compare multiple possible solutions to a problem based on how well each is likely to meet the criteria and constraints of the problem.

05 Images and Matter

Topics:

- Finding images and copyright
- Linking images and creating image tags
- Changing image size
- Atoms

Standards:

1B-AP-09: Create programs that use variables to store and modify data.

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-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-IC-21: Use public domain or creative commons media, and refrain from copying or using material created by others without permission.

NGSS 5-PS1-1: Develop a model to describe that matter is made of particles too small to be seen.

NGSS 3-5-ETS1-2: Generate and compare multiple possible solutions to a problem based on how well each is likely to meet the criteria and constraints of the problem.

06 Science Fair Project

Topics:

- Application of learned skills

Standards:

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-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-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-IC-21: Use public domain or creative commons media, and refrain from copying or using material created by others without permission.

NGSS 5-ESS3-1: Obtain and combine information about ways individual communities use science ideas to protect the Earth’s resources and environment.

NGSS 3-5-ETS1-2: Generate and compare multiple possible solutions to a problem based on how well each is likely to meet the criteria and constraints of the problem.

B Cross Curriculum: Exploring Math Through Coding

01 Sequencing in Programming and Math

Topics:

- Sequences and order
- Algorithms and how they are used in computer programming

Standards:

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.

1B-AP-15: Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.

CCSS.MATH.CONTENT.5.NBT.B.7: Add, subtract, multiply, and divide decimals to hundredths, using concrete models or drawings and strategies based on place value, properties of operations, and/or the relationship between addition and subtraction.

CCSS.MATH.PRACTICE.MP1: Make sense of problems and persevere in solving them.

CCSS.MATH.PRACTICE.MP2: Reason abstractly and quantitatively.

CCSS.MATH.PRACTICE.MP4: Model with mathematics.

CCSS.MATH.PRACTICE.MP6: Attend to precision.

CCSS.MATH.PRACTICE.MP7: Look for and make use of structure.

02 Variables in Programming and Math

Topics:

- What a variable is and its role in programming
- Dividing numbers

Standards:

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-15: Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.

CCSS.MATH.CONTENT.5.NF.B.3: Interpret a fraction as division of the numerator by the denominator (a/b = a ÷ b). Solve word problems involving division of whole numbers leading to answers in the form of fractions or mixed numbers

CCSS.MATH.PRACTICE.MP1: Make sense of problems and persevere in solving them.

CCSS.MATH.PRACTICE.MP2: Reason abstractly and quantitatively.

CCSS.MATH.PRACTICE.MP4: Model with mathematics.

CCSS.MATH.PRACTICE.MP6: Attend to precision.

CCSS.MATH.PRACTICE.MP7: Look for and make use of structure.

03 Conditionals in Programming and Math

Topics:

- Conditionals and their use in computer programming

- Inequalities comparing decimals

Standards:

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-15: Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.

CCSS.MATH.CONTENT.5.NBT.A.3: Read, write, and compare decimals to thousandths.

CCSS.MATH.PRACTICE.MP1: Make sense of problems and persevere in solving them.

CCSS.MATH.PRACTICE.MP2: Reason abstractly and quantitatively.

CCSS.MATH.PRACTICE.MP4: Model with mathematics.

CCSS.MATH.PRACTICE.MP5: Use appropriate tools strategically.

CCSS.MATH.PRACTICE.MP6: Attend to precision.

CCSS.MATH.PRACTICE.MP7: Look for and make use of structure.

CCSS.MATH.PRACTICE.MP8: Look for and express regularity in repeated reasoning.

04 HTML and Area

Topics:

- What HTML is
- HTML tags
- Area
- Multiplying fractions

Standards:

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-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-15: Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.

CCSS.MATH.CONTENT.5.NF.B.4: Apply and extend previous understandings of multiplication to multiply a fraction or whole number by a fraction.

CCSS.MATH.PRACTICE.MP1: Make sense of problems and persevere in solving them.

CCSS.MATH.PRACTICE.MP2: Reason abstractly and quantitatively.

CCSS.MATH.PRACTICE.MP4: Model with mathematics.

CCSS.MATH.PRACTICE.MP5: Use appropriate tools strategically.

CCSS.MATH.PRACTICE.MP6: Attend to precision.

CCSS.MATH.PRACTICE.MP7: Look for and make use of structure.

05 Images and Geometry

Topics:

- Finding images and copyright
- Linking images and creating image tags
- Changing Image Size
- Geometry

Standards:

1B-AP-09: Create programs that use variables to store and modify data.

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-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-IC-21: Use public domain or creative commons media, and refrain from copying or using material created by others without permission.

CCSS.MATH.CONTENT.5.G.B.3: Understand that attributes belonging to a category of two-dimensional figures also belong to all subcategories of that category.

CCSS.MATH.PRACTICE.MP1: Make sense of problems and persevere in solving them.

CCSS.MATH.PRACTICE.MP2: Reason abstractly and quantitatively.

CCSS.MATH.PRACTICE.MP4: Model with mathematics.

CCSS.MATH.PRACTICE.MP5: Use appropriate tools strategically.

CCSS.MATH.PRACTICE.MP6: Attend to precision.

CCSS.MATH.PRACTICE.MP7: Look for and make use of structure.

C Cross Curriculum: Exploring English Language Arts Through Coding

01 Headers and Subtitles

Topics:

- Header tags
- Header hierarchy
- Writing subtitles

Standards:

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-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-15: Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.

CCSS.ELA-LITERACY.W.5.2.A: Introduce a topic clearly, provide a general observation and focus, and group related information logically; include formatting (e.g., headings), illustrations, and multimedia when useful to aiding comprehension.

02 Paragraphs

Topics:

- Paragraph tags
- Writing hook sentences
- Writing introduction paragraphs
- Writing body paragraphs
- Writing conclusion paragraphs

Standards:

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-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-15: Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.

CCSS.ELA-LITERACY.W.5.1.A: Introduce a topic or text clearly, state an opinion, and create an organizational structure in which ideas are logically grouped to support the writer's purpose.

CCSS.ELA-LITERACY.W.5.1.B: Provide logically ordered reasons that are supported by facts and details.

CCSS.ELA-LITERACY.W.5.2.A: Introduce a topic clearly, provide a general observation and focus, and group related information logically; include formatting (e.g., headings), illustrations, and multimedia when useful to aiding comprehension.

CCSS.ELA-LITERACY.W.5.4: Produce clear and coherent writing in which the development and organization are appropriate to task, purpose, and audience.

CCSS.ELA-LITERACY.W.5.10: Write routinely over extended time frames and shorter time frames for a range of discipline-specific tasks, purposes, and audiences.

03 Opinion Writing

Topics:

- Application of learned skills

- Writing practice

Standards:

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-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-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-IC-21: Use public domain or creative commons media, and refrain from copying or using material created by others without permission.

CCSS.ELA-LITERACY.W.5.1: Write opinion pieces on topics or texts, supporting a point of view with reasons and information.

CCSS.ELA-LITERACY.W.5.4: Produce clear and coherent writing in which the development and organization are appropriate to task, purpose, and audience.

CCSS.ELA-LITERACY.W.5.5: With guidance and support from peers and adults, develop and strengthen writing as needed by planning, revising, editing, rewriting, or trying a new approach.

CCSS.ELA-LITERACY.W.5.6: With some guidance and support from adults, use technology, including the Internet, to produce and publish writing as well as to interact and collaborate with others; demonstrate sufficient command of keyboarding skills to type a minimum of two pages in a single sitting.

CCSS.ELA-LITERACY.W.5.10: Write routinely over extended time frames and shorter time frames for a range of discipline-specific tasks, purposes, and audiences.

CCSS.ELA-LITERACY.L.5.1: Demonstrate command of the conventions of standard English grammar and usage when writing or speaking.

CCSS.ELA-LITERACY.L.5.2: Demonstrate command of the conventions of standard English capitalization, punctuation, and spelling when writing.

04 Narrative Writing

Topics:

- Application of learned skills
- Writing practice

Standards:

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-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-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-IC-21: Use public domain or creative commons media, and refrain from copying or using material created by others without permission.

CCSS.ELA-LITERACY.W.5.3: Write narratives to develop real or imagined experiences or events using effective technique, descriptive details, and clear event sequences.

CCSS.ELA-LITERACY.W.5.4: Produce clear and coherent writing in which the development and organization are appropriate to task, purpose, and audience.

CCSS.ELA-LITERACY.W.5.5: With guidance and support from peers and adults, develop and strengthen writing as needed by planning, revising, editing, rewriting, or trying a new approach.

CCSS.ELA-LITERACY.W.5.6: With some guidance and support from adults, use technology, including the Internet, to produce and publish writing as well as to interact and collaborate with others; demonstrate sufficient command of keyboarding skills to type a minimum of two pages in a single sitting.

CCSS.ELA-LITERACY.W.5.10: Write routinely over extended time frames and shorter time frames for a range of discipline-specific tasks, purposes, and audiences.

CCSS.ELA-LITERACY.L.5.1: Demonstrate command of the conventions of standard English grammar and usage when writing or speaking.

CCSS.ELA-LITERACY.L.5.2: Demonstrate command of the conventions of standard English capitalization, punctuation, and spelling when writing.

05 Informational Writing

Topics:

- Application of learned skills
- Writing practice

Standards:

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-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-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-IC-21: Use public domain or creative commons media, and refrain from copying or using material created by others without permission.

CCSS.ELA-LITERACY.W.5.2: Write informative/explanatory texts to examine a topic and convey ideas and information clearly.

CCSS.ELA-LITERACY.W.5.4: Produce clear and coherent writing in which the development and organization are appropriate to task, purpose, and audience.

CCSS.ELA-LITERACY.W.5.5: With guidance and support from peers and adults, develop and strengthen writing as needed by planning, revising, editing, rewriting, or trying a new approach.

CCSS.ELA-LITERACY.W.5.6: With some guidance and support from adults, use technology, including the Internet, to produce and publish writing as well as to interact and collaborate with others; demonstrate sufficient command of keyboarding skills to type a minimum of two pages in a single sitting.

CCSS.ELA-LITERACY.W.5.7: Conduct short research projects that use several sources to build knowledge through investigation of different aspects of a topic.

CCSS.ELA-LITERACY.W.5.8: Recall relevant information from experiences or gather relevant information from print and digital sources; summarize or paraphrase information in notes and finished work, and provide a list of sources.

CCSS.ELA-LITERACY.W.5.9: Draw evidence from literary or informational texts to support analysis, reflection, and research.

CCSS.ELA-LITERACY.W.5.10: Write routinely over extended time frames and shorter time frames for a range of discipline-specific tasks, purposes, and audiences.

CCSS.ELA-LITERACY.L.5.1: Demonstrate command of the conventions of standard English grammar and usage when writing or speaking.

CCSS.ELA-LITERACY.L.5.2: Demonstrate command of the conventions of standard English capitalization, punctuation, and spelling when writing.

Digital Citizenship

01 Digital Footprint

Topics:

- Creating a positive digital footprint

Standards:

1A-IC-16: Compare how people live and work before and after the implementation or adoption of new computing technology.

1B-IC-18: Discuss computing technologies that have changed the world, and express how those technologies influence, and are influenced by, cultural practices.

02 Internet Safety

Topics:

- Don't talk to strangers
- Don't share personal information
- Beware of scams

Standards:

1A-IC-18: Keep login information private, and log off of devices appropriately.

1B-NI-05: Discuss real-world cybersecurity problems and how personal information can be protected.

03 Cybersecurity

Topics:

- Passwords
- Log out of computers
- Keep personal information private
- Beware of what you download

Standards:

1A-NI-04: Model how information is broken down into smaller pieces, transmitted as packets through multiple devices over networks and the Internet, and reassembled at the destination.

1A-IC-18: Keep login information private, and log off of devices appropriately.

1B-NI-05: Discuss real-world cybersecurity problems and how personal information can be protected.

04 Cyberbullying

Topics:

- Be responsible and respectful
- How to respond to cyberbullying

Standards:

1A-IC-17: Work respectfully and responsibly with others online.

1B-IC-20: Seek diverse perspectives for the purpose of improving computational artifacts.

05 Giving Credit to Others

Topics:

- Appropriate Attribution
- Property Rights

Standards:

1A-IC-17: Work respectfully and responsibly with others online.

1A-AP-13: Give attribution when using the ideas and creations of others while developing programs.

1B-AP-14: Observe intellectual property rights and give appropriate attribution when creating or remixing programs.

06 Digital Health and Wellness

Topics:

- Finding Balance between Plugged and Unplugged Activities

Standards:

None.

Skill
Struck ®

HTML 1/ CSS 1

Course Overview and Goals

The HTML/CSS course teaches the basics of building a website. Students will practice creating elements on a page and polishing them with a variety of styling techniques. Through projects and challenges, students will exercise logical thinking processes and critical thinking skills. By the end of the course, students will be able to build a basic website using HTML and CSS.

Who is it for?

The JavaScript course is designed for beginning JavaScript students. No previous programming experience is required.

01 What is a Computer?

Topics:

- Computer Parts and Pieces

Platform activities:

- HTML Basics
- Headers
- Text Color
- Fonts

Standards:

2-CS-02: Design projects that combine hardware and software components to collect and exchange data.

2-DA-07: Represent data using multiple encoding schemes.

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

2-AP-16: Incorporate existing code, media, and libraries into original programs, and give attribution.

02 Digital Safety and Citizenship

Topics:

- Digital Citizenship
- Digital Safety and Information Security

Platform activities:

- HTML as a Language
- HTML Structure
- Background Color
- Title

Standards:

2-CS-01: Recommend improvements to the design of computing devices, based on an analysis of how users interact with the devices.

2-CS-02: Design projects that combine hardware and software components to collect and exchange data.

2-NI-05: Explain how physical and digital security measures protect electronic information.

2-NI-06: Apply multiple methods of encryption to model the secure transmission of information.

2-DA-08: Collect data using computational tools and transform the data to make it more useful and reliable.

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

2-AP-16: Incorporate existing code, media, and libraries into original programs, and give attribution.

2-IC-20: Compare tradeoffs associated with computing technologies that affect people's everyday activities and career options.

2-IC-23: Describe tradeoffs between allowing information to be public and keeping information private and secure.

03 How the Internet Works

Topics:

- Packets and the Internet
- How the Internet Works

Platform activities:

- Paragraphs

- Breaks & Dividers
- Ordered Lists
- Unordered Lists
- Text Align
- Links

Standards:

2-CS-01: Recommend improvements to the design of computing devices, based on an analysis of how users interact with the devices.

2-CS-02: Design projects that combine hardware and software components to collect and exchange data.

2-NI-04: Model the role of protocols in transmitting data across networks and the Internet.

2-NI-05: Explain how physical and digital security measures protect electronic information.

2-DA-07: Represent data using multiple encoding schemes.

2-DA-08: Collect data using computational tools and transform the data to make it more useful and reliable.

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

2-AP-16: Incorporate existing code, media, and libraries into original programs, and give attribution.

2-IC-23: Describe tradeoffs between allowing information to be public and keeping information private and secure.

04 Computers and Your World

Topics:

- Computers and Culture

Platform activities:

- Button
- Images
- Background Images
- Videos

Standards:

2-DA-07: Represent data using multiple encoding schemes.

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

2-AP-16: Incorporate existing code, media, and libraries into original programs, and give attribution.

2-AP-17: Systematically test and refine programs using a range of test cases.

2-IC-20: Compare tradeoffs associated with computing technologies that affect people's everyday activities and career options.

2-IC-21: Discuss issues of bias and accessibility in the design of existing technologies.

05 Problem Solving Skills

Topics:

- Problem Solving Skills Practice

Platform activities:

- Links
- Marquees

Standards:

2-CS-03: Systematically identify and fix problems with computing devices and their components.

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

2-AP-13: Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.

2-AP-15: Seek and incorporate feedback from team members and users to refine a solution that meets user needs.

06 Team Collaboration

Topics:

- Working on a team
- Peer Feedback

Platform activities:

- Team Project

Standards:

2-CS-01: Recommend improvements to the design of computing devices, based on an analysis of how users interact with the devices.

2-CS-02: Design projects that combine hardware and software components to collect and exchange data.

2-DA-07: Represent data using multiple encoding schemes.

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

2-AP-13: Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.

2-AP-15: Seek and incorporate feedback from team members and users to refine a solution that meets user needs.

2-AP-16: Incorporate existing code, media, and libraries into original programs, and give attribution.

2-AP-17: Systematically test and refine programs using a range of test cases.

2-AP-18: Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.

2-AP-19: Document programs in order to make them easier to follow, test, and debug.

2-IC-22: Collaborate with many contributors through strategies such as crowdsourcing or surveys when creating a computational artifact.

07 Careers in Web Development

Topics:

- Exploration of Career Opportunities

Platform activities:

- The Style Tag
- CSS Intro
- Text Shadow,
- Background Image Style

Standards:

2-DA-07: Represent data using multiple encoding schemes.

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

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.

2-IC-20: Compare tradeoffs associated with computing technologies that affect people's everyday activities and career options.

08 Communication in a Digital World

Topics:

- Documenting Code
- Etiquette on Digital Platforms

Platform activities:

- External Style Sheet
- Comments
- Classes
- Div Tag

Standards:

2-CS-01: Recommend improvements to the design of computing devices, based on an analysis of how users interact with the devices.

2-CS-02: Design projects that combine hardware and software components to collect and exchange data.

2-NI-06: Apply multiple methods of encryption to model the secure transmission of information.

2-DA-07: Represent data using multiple encoding schemes.

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

2-AP-13: Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.

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.

2-AP-17: Systematically test and refine programs using a range of test cases.

2-AP-19: Document programs in order to make them easier to follow, test, and debug.

2-IC-20: Compare tradeoffs associated with computing technologies that affect people's everyday activities and career options.

09 Accessibility

Topics:

- Exploration of Improving Accessibility
- Improving Accessibility of Computer Programs

Platform activities:

- Margins and Padding
- Color Picker
- Color Gradient
- CSS Icons

Standards:

2-CS-01: Recommend improvements to the design of computing devices, based on an analysis of how users interact with the devices.

2-CS-02: Design projects that combine hardware and software components to collect and exchange data.

2-NI-06: Apply multiple methods of encryption to model the secure transmission of information.

2-DA-07: Represent data using multiple encoding schemes.

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

2-AP-16: Incorporate existing code, media, and libraries into original programs, and give attribution.

2-IC-20: Compare tradeoffs associated with computing technologies that affect people's everyday activities and career options.

2-IC-21: Discuss issues of bias and accessibility in the design of existing technologies.

10 Troubleshooting

Topics:

- Computer Troubleshooting Practice

Platform activities:

- List Styles/ Rounded Corners
- Borders
- Box Shadow

Standards:

2-CS-02: Design projects that combine hardware and software components to collect and exchange data.

2-CS-03: Systematically identify and fix problems with computing devices and their components.

2-DA-07: Represent data using multiple encoding schemes.

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

2-AP-13: Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.

2-AP-16: Incorporate existing code, media, and libraries into original programs, and give attribution.

2-AP-19: Document programs in order to make them easier to follow, test, and debug.

2-IC-21: Discuss issues of bias and accessibility in the design of existing technologies.

11 The Design Process

Topics:

- Basic Principles of Design
- Updating Websites

Platform activities:

- Button Design
- Pseudo Selectors
- Opacity

Standards:

2-CS-01: Recommend improvements to the design of computing devices, based on an analysis of how users interact with the devices.

2-CS-03: Systematically identify and fix problems with computing devices and their components.

2-DA-07: Represent data using multiple encoding schemes.

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

2-AP-13: Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.

2-AP-16: Incorporate existing code, media, and libraries into original programs, and give attribution.

2-AP-17: Systematically test and refine programs using a range of test cases.

12 Encryption & Flowcharts

Topics:

- Using Flowcharts
- Encryption Practice

Platform activities:

- Positioning
- Media Queries

Standards:

2-CS-03: Systematically identify and fix problems with computing devices and their components.

2-NI-06: Apply multiple methods of encryption to model the secure transmission of information.

2-DA-07: Represent data using multiple encoding schemes.

2-AP-10: Use flowcharts and/or pseudocode to address complex problems as algorithms.

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

2-AP-19: Document programs in order to make them easier to follow, test, and debug.

13 Market Research

Topics:

- Collecting and Representing Data
- Analyzing Advertising
- Collecting Feedback

Standards:

2-CS-01: Recommend improvements to the design of computing devices, based on an analysis of how users interact with the devices.

2-DA-07: Represent data using multiple encoding schemes.

2-DA-08: Collect data using computational tools and transform the data to make it more useful and reliable.

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

2-AP-15: Seek and incorporate feedback from team members and users to refine a solution that meets user needs.

2-IC-20: Compare tradeoffs associated with computing technologies that affect people's everyday activities and career options.

2-IC-22: Collaborate with many contributors through strategies such as crowdsourcing or surveys when creating a computational artifact.

2-IC-23: Describe tradeoffs between allowing information to be public and keeping information private and secure.

14 Working on a Team

Topics:

- Team Dynamics
- Teamwork Skills

- Project Planning
- Time Management

Standards:

2-CS-01: Recommend improvements to the design of computing devices, based on an analysis of how users interact with the devices.

2-DA-07: Represent data using multiple encoding schemes.

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

2-AP-13: Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.

2-AP-14: Create procedures with parameters to organize code and make it easier to reuse.

2-AP-15: Seek and incorporate feedback from team members and users to refine a solution that meets user needs.

2-AP-16: Incorporate existing code, media, and libraries into original programs, and give attribution.

2-AP-17: Systematically test and refine programs using a range of test cases.

2-AP-18: Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.

2-AP-19: Document programs in order to make them easier to follow, test, and debug.

2-IC-22: Collaborate with many contributors through strategies such as crowdsourcing or surveys when creating a computational artifact.

15 Working on a Team & Market Research

Topics:

- Team Dynamics
- Teamwork Skills
- Project Planning
- Time Management

Standards:

2-CS-01: Recommend improvements to the design of computing devices, based on an analysis of how users interact with the devices.

2-DA-07: Represent data using multiple encoding schemes.

2-DA-09: Refine computational models based on the data they have generated.

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

2-AP-13: Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.

2-AP-14: Create procedures with parameters to organize code and make it easier to reuse.

2-AP-15: Seek and incorporate feedback from team members and users to refine a solution that meets user needs.

2-AP-16: Incorporate existing code, media, and libraries into original programs, and give attribution.

2-AP-17: Systematically test and refine programs using a range of test cases.

2-AP-18: Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.

2-AP-19: Document programs in order to make them easier to follow, test, and debug.

2-IC-22: Collaborate with many contributors through strategies such as crowdsourcing or surveys when creating a computational artifact.

16 Presenting Code

Topics:

- Reviewing concepts
- Reflection on Learning
- Future Learning Opportunities

Standards:

2-CS-01: Recommend improvements to the design of computing devices, based on an analysis of how users interact with the devices.

2-DA-07: Represent data using multiple encoding schemes.

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

2-AP-13: Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.

2-AP-14: Create procedures with parameters to organize code and make it easier to reuse.

2-AP-15: Seek and incorporate feedback from team members and users to refine a solution that meets user needs.

2-AP-16: Incorporate existing code, media, and libraries into original programs, and give attribution.

2-AP-17: Systematically test and refine programs using a range of test cases.

2-AP-18: Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.

2-AP-19: Document programs in order to make them easier to follow, test, and debug.

2-IC-20: Compare tradeoffs associated with computing technologies that affect people's everyday activities and career options.

2-IC-21: Discuss issues of bias and accessibility in the design of existing technologies.

2-IC-22: Collaborate with many contributors through strategies such as crowdsourcing or surveys when creating a computational artifact.

Skill
Struck ®

HTML 2 / CSS 2

Course Overview and Goals

This course dives deeper into web development as students discover more of what HTML and CSS can do. Students explore the possibilities of bootstrap, animations, and tables just to name a few topics. Students polish their websites further as they have more sophistication and interaction to their sites. Students will need to have completed HTML 1 and CSS 1 to understand HTML 2 & CSS 2.

Who is it for?

The HTML 2/ CSS 2 course is designed for students who have completed HTML 1 / CSS 1. Some previous programming experience is required.

01 Bootstrap Intro

Topics:

- Bootstrap

Standards:

2-DA-07: Represent data using multiple encoding schemes.

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

3A-AP-18: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.

3A-AP-23: Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.

3B-AP-16: Demonstrate code reuse by creating programming solutions using libraries and APIs.

02 Bootstrap Classes

Topics:

- Bootstrap Classes

Standards:

2-DA-07: Represent data using multiple encoding schemes.

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

3A-AP-18: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.

3A-AP-23: Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.

3B-AP-16: Demonstrate code reuse by creating programming solutions using libraries and APIs.

03 Bootstrap Grid

Topics:

- Bootstrap Grid

Standards:

2-DA-07: Represent data using multiple encoding schemes.

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

3A-AP-18: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.

3A-AP-23: Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.

3B-AP-16: Demonstrate code reuse by creating programming solutions using libraries and APIs.

04 Twelve Column Grid

Topics:

- Twelve column grid

Standards:

2-DA-07: Represent data using multiple encoding schemes.

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

3A-AP-18: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.

3A-AP-23: Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.

3B-AP-16: Demonstrate code reuse by creating programming solutions using libraries and APIs.

05 Jumbotron

Topics:

- Jumbotron

Standards:

2-DA-07: Represent data using multiple encoding schemes.

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

3A-AP-18: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.

3A-AP-23: Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.

3B-AP-16: Demonstrate code reuse by creating programming solutions using libraries and APIs.

06 Cards

Topics:

- Cards

Standards:

2-DA-07: Represent data using multiple encoding schemes.

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

3A-AP-18: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.

3A-AP-23: Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.

3B-AP-16: Demonstrate code reuse by creating programming solutions using libraries and APIs.

07 Carousels

Topics:

- Carousels

Standards:

2-DA-07: Represent data using multiple encoding schemes.

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

3A-AP-18: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.

3A-AP-23: Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.

3B-AP-16: Demonstrate code reuse by creating programming solutions using libraries and APIs.

08 Bootstrap Nav Bar

Topics:

- Bootstrap nav bar

Standards:

2-DA-07: Represent data using multiple encoding schemes.

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

3A-AP-18: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.

3A-AP-23: Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.

3B-AP-16: Demonstrate code reuse by creating programming solutions using libraries and APIs.

09 Animations

Topics:

- Animations

Standards:

2-DA-07: Represent data using multiple encoding schemes.

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

3A-AP-18: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.

3A-AP-23: Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.

3B-AP-16: Demonstrate code reuse by creating programming solutions using libraries and APIs.

10 Bootstrap Dropdown

Topics:

- Bootstrap dropdown

Standards:

2-DA-07: Represent data using multiple encoding schemes.

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

3A-AP-18: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.

3A-AP-23: Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.

3B-AP-16: Demonstrate code reuse by creating programming solutions using libraries and APIs.

11 Transform

Topics:

- Transform

Standards:

2-DA-07: Represent data using multiple encoding schemes.

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

3A-AP-18: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.

3A-AP-23: Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.

12 Transition

Topics:

- Transition

Standards:

2-DA-07: Represent data using multiple encoding schemes.

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

3A-AP-18: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.

3A-AP-23: Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.

13 Tables Intro

Topics:

- Tables intro

Standards:

2-DA-07: Represent data using multiple encoding schemes.

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

3A-AP-18: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.

3A-AP-23: Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.

14 Tables Continued

Topics:

- Tables continued

Standards:

2-DA-07: Represent data using multiple encoding schemes.

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

3A-AP-18: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.

3A-AP-23: Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.

15 Table Styles

Topics:

- Table Styles

Standards:

2-DA-07: Represent data using multiple encoding schemes.

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

3A-AP-18: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.

3A-AP-23: Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.

16 Forms

Topics:

- Forms

Standards:

2-DA-07: Represent data using multiple encoding schemes.

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

3A-AP-18: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.

3A-AP-23: Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.

17 Text Input

Topics:

- Text Input

Standards:

2-DA-07: Represent data using multiple encoding schemes.

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

3A-AP-18: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.

3A-AP-23: Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.

18 Radio Input

Topics:

- Radio Input

Standards:

2-DA-07: Represent data using multiple encoding schemes.

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

3A-AP-18: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.

3A-AP-23: Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.

19 Submit Input

Topics:

- Submit Input

Standards:

2-DA-07: Represent data using multiple encoding schemes.

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

3A-AP-18: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.

3A-AP-23: Document design decisions using text, graphics, presentations, and/or demonstrations in the development of complex programs.

Skill Struck®

JavaScript 1

Course Overview and Goals

The JavaScript course teaches the basics of the JavaScript language. Students will practice algorithmic thinking and critical problem solving as it applies to computer programming. Through projects and challenges, students will exercise logical thinking processes. By the end of the course, students will be able to complete basic programming in the JavaScript language.

Who is it for?

The JavaScript course is designed for beginning JavaScript students. No previous programming experience is required.

01 What is Computer Programming?

Topics:

- Overview of Computer Programming
- Computer Programming and Careers
- Password Strength and Safety
- Computer Safety

Standards:

2-IC-20: Compare tradeoffs associated with computing technologies that affect people’s everyday activities and career options.

2-NI-05: Explain how physical and digital security measures protect electronic information

02 Digital Citizenship

Topics:

- How to be a Good Digital Citizen
- Digital Footprints
- Social Media

Platform activities:

- Introduction to the Voyage Platform
- JavaScript Intro
- Console Logs

Standards:

2-IC-20: Compare tradeoffs associated with computing technologies that affect people’s everyday activities and career options.

2-NI-05: Explain how physical and digital security measures protect electronic information.

2-IC-23: Describe tradeoffs between allowing information to be public and keeping information private and secure.

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

2-AP-17: Systematically test and refine programs using a range of test cases

03 How the Internet Works

Topics:

- Internet Protocols and Packets
- Internet Hardware

Platform activities:

- JavaScript Math

Standards:

2-NI-04: Model the role of protocols in transmitting data across networks and the Internet.

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

2-AP-17: Systematically test and refine programs using a range of test cases.

04 Programming and Your World

Topics:

- Internet Cookies
- Privacy in a Digital Society
- Phishing

Platform activities:

- Concatenation

Standards:

2-IC-20: Compare tradeoffs associated with computing technologies that affect people’s everyday activities and career options.

2-NI-05: Explain how physical and digital security measures protect electronic information.

2-IC-23: Describe tradeoffs between allowing information to be public and keeping information private and secure.

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

2-AP-17: Systematically test and refine programs using a range of test cases

05 Problem Solving

Topics:

- Critical Thinking Skills
- Problem Solving Strategies
- Encryption

Platform activities:

- Functions
- Parameters

Standards:

2-AP-13: Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

2-AP-17: Systematically test and refine programs using a range of test cases.

2-AP-14: Create procedures with parameters to organize code and make it easier to reuse.

2-NI-06: Apply multiple methods of encryption to model the secure transmission of information.

2-AP-15: Seek and incorporate feedback from team members and users to refine a solution that meets user needs.

06 Data Representation

Topics:

- Different Methods of Data Representation
- Data Collection

Platform activities:

- Return Statement
- Commenting in JavaScript

Standards:

2-DA-08: Collect data using computational tools and transform the data to make it more useful and reliable.

2-DA-07: Represent data using multiple encoding schemes.

2-AP-19: Document programs in order to make them easier to follow, test, and debug.

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

2-AP-17: Systematically test and refine programs using a range of test cases.

07 Careers in Web Development

Topics:

- Exploration of Web Development Careers
- Conditionals

Platform activities

- If Statements
- If Statement Conditions

Standards:

2-IC-20: Compare tradeoffs associated with computing technologies that affect people's everyday activities and career options.

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

2-AP-17: Systematically test and refine programs using a range of test cases.

2-AP-12: Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.

08 Breaking Down Projects into Tasks

Topics:

- Strategies to Break Down Projects
- Problem Solving Strategies

Platform activities:

- Else If Statements

Standards:

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

2-AP-17: Systematically test and refine programs using a range of test cases.

2-AP-12: Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.

2-AP-13: Decompose problems and subproblems into parts to facilitate the design, implementation, and review of problems.

2-AP-15: Seek and incorporate feedback from team members and users to refine a solution that meets user needs.

09 Accessibility

Topics:

- Digital Accessibility
- Recommendations for Accessibility

Platform activities

- Arrays Intro
- Accessing Elements in an Array
- Array Length

Standards:

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

2-AP-17: Systematically test and refine programs using a range of test cases.

2-AP-13: Decompose problems and subproblems into parts to facilitate the design, implementation, and review of problems.

2-CS-01: Recommend improvements to the design of computing devices, based on an analysis of how users interact with the devices.

2-CS-02: Design projects that combine hardware and software components to collect and exchange data.

10 Troubleshooting

Topics:

- Computer Parts and Pieces

- Troubleshooting Strategies

Platform activities:

- Adding to Arrays
- Removing from Arrays
- Inserting Into Arrays
- Changing Arrays

Standards:

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

2-AP-17: Systematically test and refine programs using a range of test cases.

2-AP-13: Decompose problems and subproblems into parts to facilitate the design, implementation, and review of problems.

2-CS-03: Systematically identify and fix problems with computing devices and their components.

11 Flow Charts & Algorithms

Topics:

- Algorithm Practice
- Flow Charts

Platform activities:

- For Loops

Standards:

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

2-AP-13: Decompose problems and subproblems into parts to facilitate the design, implementation, and review of problems.

2-AP-10: Use flowcharts and/or pseudocode to address complex problems as algorithms.

2-AP-15: Seek and incorporate feedback from team members and users to refine a solution that meets user needs.

2-AP-12: Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.

12 Encryption

Topics:

- Binary
- Looping

Platform activities:

- For Loops

Standards:

2-DA-07: Represent data using multiple encoding schemes.

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

2-AP-13: Decompose problems and subproblems into parts to facilitate the design, implementation, and review of problems.

2-AP-12: Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.

2-AP-15: Seek and incorporate feedback from team members and users to refine a solution that meets user needs.

13 Technology and Society

Topics:

- Computers and Your World
- Artificial Intelligence
- Automation

Platform activities:

- Nested Loops

Standards:

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

2-AP-13: Decompose problems and subproblems into parts to facilitate the design, implementation, and review of problems.

2-AP-12: Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.

2-IC-20: Compare tradeoffs associated with computing technologies that affect people's everyday activities and career options.

2-AP-15: Seek and incorporate feedback from team members and users to refine a solution that meets user needs.

2-IC-22: Collaborate with many contributors through strategies such as crowdsourcing or surveys when creating a computational artifact.

14 Working on a Team

Topics:

- Teamwork Skills
- Setting up the Team Project
- Time Management
- Loops and Arrays

Standards:

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

2-AP-13: Decompose problems and subproblems into parts to facilitate the design, implementation, and review of problems.

2-AP-12: Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.

2-IC-20: Compare tradeoffs associated with computing technologies that affect people’s everyday activities and career options.

2-AP-15: Seek and incorporate feedback from team members and users to refine a solution that meets user needs.

2-IC-22: Collaborate with many contributors through strategies such as crowdsourcing or surveys when creating a computational artifact.

2-AP-18: Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.

15 Team Strategy

Topics:

- Teamwork Skills
- Working with Others
- Time Management

Standards:

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

2-AP-13: Decompose problems and subproblems into parts to facilitate the design, implementation, and review of problems.

2-AP-15: Seek and incorporate feedback from team members and users to refine a solution that meets user needs.

2-IC-22: Collaborate with many contributors through strategies such as crowdsourcing or surveys when creating a computational artifact.

2-AP-18: Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.

2-AP-19: Document programs in order to make them easier to follow, test, and debug.

16 Presenting Code

Topics:

- Reviewing concepts
- Reflection on Learning
- Future Learning Opportunities

Standards:

2-AP-15: Seek and incorporate feedback from team members and users to refine a solution that meets user needs.

2-AP-18: Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.

2-AP-19: Document programs in order to make them easier to follow, test, and debug.

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

2-AP-13: Decompose problems and subproblems into parts to facilitate the design, implementation, and review of problems.

2-IC-20: Compare tradeoffs associated with computing technologies that affect people's everyday activities and career options.

JavaScript 2

Course Overview and Goals

In this course students dive deeper into more advanced JavaScript topics as they learn about objects and the basics of object oriented programming. They discover the possibilities of jQuery and learn just how dynamic their websites can be. Students will need to have completed HTML 1 in addition to JavaScript 1 in order to understand JavaScript 2.

Who is it for?

The JavaScript 2 course is designed for students who have already completed JavaScript 1. Some previous programming experience is required.

01 Intro to Objects

Topics:

- JavaScript objects

Standards:

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

2-AP-13: Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.

3A-CS-01: Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects.

3A-AP-14: Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables.

3A-AP-17: Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.

3A-AP-18: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.

3B-AP-10: Use and adapt classic algorithms to solve computational problems.

3B-AP-11: Evaluate algorithms in terms of their efficiency, correctness, and clarity.

3B-AP-12: Compare and contrast fundamental data structures and their uses.

3B-AP-14: Construct solutions to problems using student-created components, such as procedures, modules and/or objects.

2-AP-17: Systematically test and refine programs using a range of test cases.

02 Declaring an Objects

Topics:

- Declaring JavaScript objects

Standards:

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

2-AP-13: Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.

3A-CS-01: Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects.

3A-AP-14: Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables.

3A-AP-17: Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.

3A-AP-18: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.

3B-AP-10: Use and adapt classic algorithms to solve computational problems.

3B-AP-11: Evaluate algorithms in terms of their efficiency, correctness, and clarity.

3B-AP-12: Compare and contrast fundamental data structures and their uses.

3B-AP-14: Construct solutions to problems using student-created components, such

as procedures, modules and/or objects.

2-AP-17: Systematically test and refine programs using a range of test cases.

03 Accessing Object Attributes / Properties

Topics:

- Accessing Object Attributes / Properties

Standards:

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

2-AP-13: Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.

3A-CS-01: Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects.

3A-AP-14: Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables.

3A-AP-17: Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.

3A-AP-18: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.

3B-AP-10: Use and adapt classic algorithms to solve computational problems.

3B-AP-11: Evaluate algorithms in terms of their efficiency, correctness, and clarity.

3B-AP-12: Compare and contrast fundamental data structures and their uses.

3B-AP-14: Construct solutions to problems using student-created components, such as procedures, modules and/or objects.

2-AP-17: Systematically test and refine programs using a range of test cases.

04 Methods

Topics:

- Methods

Standards:

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

2-AP-13: Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.

3A-CS-01: Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects.

3A-AP-14: Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables.

3A-AP-17: Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.

3A-AP-18: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.

3B-AP-10: Use and adapt classic algorithms to solve computational problems.

3B-AP-11: Evaluate algorithms in terms of their efficiency, correctness, and clarity.

3B-AP-12: Compare and contrast fundamental data structures and their uses.

3B-AP-14: Construct solutions to problems using student-created components, such as procedures, modules and/or objects.

2-AP-17: Systematically test and refine programs using a range of test cases.

05 Constructors

Topics:

- Constructors

Standards:

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

2-AP-13: Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.

3A-CS-01: Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects.

3A-AP-14: Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables.

3A-AP-17: Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.

3A-AP-18: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.

3B-AP-10: Use and adapt classic algorithms to solve computational problems.

3B-AP-11: Evaluate algorithms in terms of their efficiency, correctness, and clarity.

3B-AP-12: Compare and contrast fundamental data structures and their uses.

3B-AP-14: Construct solutions to problems using student-created components, such as procedures, modules and/or objects.

2-AP-17: Systematically test and refine programs using a range of test cases.

06 Assigning Unique Names to Tags

Topics:

- Class names
- Class ID

Standards:

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

2-AP-13: Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.

3A-CS-01: Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects.

3A-AP-14: Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables.

3A-AP-17: Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.

3A-AP-18: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.

3B-AP-10: Use and adapt classic algorithms to solve computational problems.

3B-AP-11: Evaluate algorithms in terms of their efficiency, correctness, and clarity.

3B-AP-12: Compare and contrast fundamental data structures and their uses.

3B-AP-14: Construct solutions to problems using student-created components, such as procedures, modules and/or objects.

2-AP-17: Systematically test and refine programs using a range of test cases.

07 JQuery

Topics:

- JQuery

Standards:

2-AP-16: Incorporate existing code, media, and libraries into original programs, and give attribution.

3B-AP-16: Demonstrate code reuse by creating programming solutions using libraries and APIs.

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

2-AP-13: Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.

3A-CS-01: Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects.

3A-AP-14: Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables.

3A-AP-17: Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.

3A-AP-18: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.

3B-AP-10: Use and adapt classic algorithms to solve computational problems.

3B-AP-11: Evaluate algorithms in terms of their efficiency, correctness, and clarity.

3B-AP-12: Compare and contrast fundamental data structures and their uses.

3B-AP-14: Construct solutions to problems using student-created components, such

as procedures, modules and/or objects.

2-AP-17: Systematically test and refine programs using a range of test cases.

08 GetElementById

Topics:

- GetElementById

Standards:

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

2-AP-13: Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.

3A-CS-01: Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects.

3A-AP-14: Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables.

3A-AP-17: Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.

3A-AP-18: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.

3B-AP-10: Use and adapt classic algorithms to solve computational problems.

3B-AP-11: Evaluate algorithms in terms of their efficiency, correctness, and clarity.

3B-AP-12: Compare and contrast fundamental data structures and their uses.

3B-AP-14: Construct solutions to problems using student-created components, such as procedures, modules and/or objects.

2-AP-17: Systematically test and refine programs using a range of test cases.

09 Inner HTML

Topics:

- Inner HTML

Standards:

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

2-AP-13: Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.

3A-CS-01: Explain how abstractions hide the underlying implementation details of computing systems embedded in everyday objects.

3A-AP-14: Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables.

3A-AP-17: Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.

3A-AP-18: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.

3B-AP-10: Use and adapt classic algorithms to solve computational problems.

3B-AP-11: Evaluate algorithms in terms of their efficiency, correctness, and clarity.

3B-AP-12: Compare and contrast fundamental data structures and their uses.

3B-AP-14: Construct solutions to problems using student-created components, such as procedures, modules and/or objects.

2-AP-17: Systematically test and refine programs using a range of test cases.

Python 1

Course Overview and Goals

The Python course teaches the basics of the Python language. Students will practice algorithmic thinking and critical problem solving as it applies to computer programming. Through projects and challenges, students will exercise logical thinking processes. By the end of the course, students will be able to complete basic programming in the Python language.

Who is it for?

The Python course is designed for beginning Python students. No previous programming experience is required.

01 What is a Computer?

Topics:

- History of Computers
- Computer Parts and Pieces

Platform activities:

- Introduction to Python

Standards:

2-CS-02: Design projects that combine hardware and software components to collect and exchange data.

2-CS-03: Systematically identify and fix problems with computing devices and their components.

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

02 Digital Citizenship

Topics:

- What is Digital Citizenship?
- Overview of different languages
- Inputs/Outputs Discussion

Platform activities:

- Variables
- Inputs/Outputs

Standards:

2-CS-02: Design projects that combine hardware and software components to collect and exchange data.

2-CS-03: Systematically identify and fix problems with computing devices and their components.

2-NI-05: Explain how physical and digital security measures protect electronic information.

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

2-AP-17: Systematically test and refine programs using a range of test cases.

2-IC-20: Compare tradeoffs associated with computing technologies that affect people's everyday activities and career options.

03 How the Internet Works

Topics:

- How the Internet Works
- Digital Safety

Platform activities:

- Syntax & Comments
- Variable Types
- Numbers

Standards:

2-NI-04: Model the role of protocols in transmitting data across networks and the Internet.

3A-NI-05: Give examples to illustrate how sensitive data can be affected by malware and other attacks.

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

2-AP-17: Systematically test and refine programs using a range of test cases.

04 Computers and Your World

Topics:

- Computers and Your World
- Social Media
- Digital Footprint

Platform activities:

- Converting

Standards:

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

2-AP-17: Systematically test and refine programs using a range of test cases.

2-IC-20: Compare tradeoffs associated with computing technologies that affect people's everyday activities and career options.

05 Problem Solving

Topics:

- Problem Solving Strategies
- Problem Solving Experiences

Platform activities:

- Python Math

- Python Project

Standards:

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

2-AP-13: Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.

2-AP-17: Systematically test and refine programs using a range of test cases.

06 Break Down Big Projects

Topics:

- Project Deconstruction
- Break Down Projects Experience

Platform activities:

- Modulus
- Python Project

Standards:

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

2-AP-13: Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.

2-AP-17: Systematically test and refine programs using a range of test cases.

07 Careers in Software Development

Topics:

- Software Development Career Exploration
- AI, Machine Learning, Automation

Platform activities:

- Strings
- String Methods
- String Methods Continued

Standards:

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.

2-AP-17: Systematically test and refine programs using a range of test cases.

2-IC-20: Compare tradeoffs associated with computing technologies that affect people's everyday activities and career options.

08 Communication in a Digital World

Topics:

- Communication Styles
- Communication Experience
- Social Communication

Platform activities:

- Concatenation
- Python Project
- Concatenating Numbers

Standards:

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

2-AP-13: Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.

2-AP-17: Systematically test and refine programs using a range of test cases.

2-IC-20: Compare tradeoffs associated with computing technologies that affect people's everyday activities and career options.

09 Accessibility

Topics:

- Accessibility
- Digital Accessibility
- Equity

Platform activities:

- Concatenating Numbers Continued
- Python Project

Standards:

2-CS-01: Recommend improvements to the design of computing devices, based on an analysis of how users interact with the devices.

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

2-AP-13: Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.

2-AP-17: Systematically test and refine programs using a range of test cases.

2-IC-20: Compare tradeoffs associated with computing technologies that affect people's everyday activities and career options.

2-IC-21: Discuss issues of bias and accessibility in the design of existing technologies.

10 Troubleshooting

Topics:

- Troubleshooting Techniques
- Code Debugging
- Troubleshooting Exercises

Platform activities:

- Lists
- Python If Statements

Standards:

2-CS-03: Systematically identify and fix problems with computing devices and their components.

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

2-AP-13: Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.

2-AP-17: Systematically test and refine programs using a range of test cases.

11 Data Storage

Topics:

- Data Sizes
- Data Storage
- Saving Space

Platform activities:

- Python Else If Statements

Standards:

2-DA-07: Represent data using multiple encoding schemes.

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

2-AP-13: Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.

2-AP-17: Systematically test and refine programs using a range of test cases.

12 Encryption & Flowcharts

Topics:

- Encryption with Morse Code
- Ciphers
- Flow Charts for Decision Making

Platform activities:

- Python Conditionals

Standards:

2-NI-06: Apply multiple methods of encryption to model the secure transmission of information.

2-DA-07: Represent data using multiple encoding schemes.

2-AP-10: Use flowcharts and/or pseudocode to address complex problems as algorithms.

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

2-AP-13: Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.

2-AP-17: Systematically test and refine programs using a range of test cases.

13 Market Research

Topics:

- Market Research
- Surveys & Data Interpretation
- Technology and Career Shifts
- Data Privacy

Platform activities:

- While Loops

Standards:

2-DA-08: Collect data using computational tools and transform the data to make it more useful and reliable.

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

2-AP-13: Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.

2-AP-15: Seek and incorporate feedback from team members and users to refine a solution that meets user needs.

2-AP-17: Systematically test and refine programs using a range of test cases.

2-IC-22: Collaborate with many contributors through strategies such as crowdsourcing or surveys when creating a computational artifact.

2-IC-23: Describe tradeoffs between allowing information to be public and keeping information private and secure.

14 Documenting Code & Problem Solving

Topics:

- Code Documentation
- Problem Solving Experiences

Platform activities:

- Python Projects: Sentence Generator & Earnings Calculator

Standards:

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

2-AP-13: Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.

2-AP-17: Systematically test and refine programs using a range of test cases.

2-AP-19: Document programs in order to make them easier to follow, test, and debug.

15 Team Coding

Topics:

- Tips for Working on a Team
- Team Building
- Collaboration
- Feedback

Platform activities:

- Team Project

Standards:

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

2-AP-13: Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.

2-AP-15: Seek and incorporate feedback from team members and users to refine a solution that meets user needs.

2-AP-17: Systematically test and refine programs using a range of test cases.

2-AP-18: Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.

16 Reflection

Topics:

- Team Presentation
- Class Review
- Project Reflection
- Why Coding is Important

Standards

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

2-AP-13: Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.

2-AP-15: Seek and incorporate feedback from team members and users to refine a solution that meets user needs.

2-AP-17: Systematically test and refine programs using a range of test cases.

2-AP-18: Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts.

2-IC-20: Compare tradeoffs associated with computing technologies that affect people's everyday activities and career options.

Python 2

Course Overview and Goals

In this course students dive deeper into more advanced Python concepts as they learn how to manipulate lists and handle functions. They are introduced to object oriented programming concepts and practice building and using objects. Students will need to have completed Python 1 in order to understand Python 2.

Who is it for?

The Python 2 course is designed for students who have completed Python 1. Some previous programming experience is required.

01 If Statement in a For Loop

Topics:

- If Statement in a For Loop

Standards

2-AP-12: Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.

3B-AP-13: Illustrate the flow of execution of a recursive algorithm.

2-DA-07: Represent data using multiple encoding schemes.

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

2-AP-13: Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.

2-AP-17: Systematically test and refine programs using a range of test cases.

3A-AP-13: Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.

3A-AP-17: Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.

3A-AP-18: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.

3B-AP-11: Evaluate algorithms in terms of their efficiency, correctness, and clarity.

3B-AP-12: Compare and contrast fundamental data structures and their uses.

3B-AP-14: Construct solutions to problems using student-created components, such as procedures, modules and/or objects.

02 Accessing Items in a List

Topics:

- Accessing items in a list

Standards

3A-AP-14: Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables.

2-DA-07: Represent data using multiple encoding schemes.

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

2-AP-13: Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.

2-AP-17: Systematically test and refine programs using a range of test cases.

3A-AP-13: Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.

3A-AP-17: Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.

3A-AP-18: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.

3B-AP-11: Evaluate algorithms in terms of their efficiency, correctness, and clarity.

3B-AP-12: Compare and contrast fundamental data structures and their uses.

3B-AP-14: Construct solutions to problems using student-created components, such as procedures, modules and/or objects.

Skill Struck ®

03 Using Integers from a List

Topics:

- Using Integers from a List

Standards

3A-AP-14: Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables.

2-DA-07: Represent data using multiple encoding schemes.

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

2-AP-13: Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.

2-AP-17: Systematically test and refine programs using a range of test cases.

3A-AP-13: Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.

3A-AP-17: Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.

3A-AP-18: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.

3B-AP-11: Evaluate algorithms in terms of their efficiency, correctness, and clarity.

3B-AP-12: Compare and contrast fundamental data structures and their uses.

3B-AP-14: Construct solutions to problems using student-created components, such as procedures, modules and/or objects.

04 Adding to Lists in Python

Topics:

- Adding to lists

Standards

3A-AP-14: Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables.

2-DA-07: Represent data using multiple encoding schemes.

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

2-AP-13: Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.

2-AP-17: Systematically test and refine programs using a range of test cases.

3A-AP-13: Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.

3A-AP-17: Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.

3A-AP-18: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.

3B-AP-11: Evaluate algorithms in terms of their efficiency, correctness, and clarity.

3B-AP-12: Compare and contrast fundamental data structures and their uses.

3B-AP-14: Construct solutions to problems using student-created components, such as procedures, modules and/or objects.

05 Removing Lists in Python

Topics:

- Removing lists

Standards

3A-AP-14: Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables.

2-DA-07: Represent data using multiple encoding schemes.

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

2-AP-13: Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.

2-AP-17: Systematically test and refine programs using a range of test cases.

3A-AP-13: Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.

3A-AP-17: Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.

3A-AP-18: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.

3B-AP-11: Evaluate algorithms in terms of their efficiency, correctness, and clarity.

3B-AP-12: Compare and contrast fundamental data structures and their uses.

3B-AP-14: Construct solutions to problems using student-created components, such as procedures, modules and/or objects.

06 Python Lists Continued

Topics:

- Python lists

Standards

3A-AP-14: Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables.

2-DA-07: Represent data using multiple encoding schemes.

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

2-AP-13: Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.

2-AP-17: Systematically test and refine programs using a range of test cases.

3A-AP-13: Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.

3A-AP-17: Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.

3A-AP-18: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.

3B-AP-11: Evaluate algorithms in terms of their efficiency, correctness, and clarity.

3B-AP-12: Compare and contrast fundamental data structures and their uses.

3B-AP-14: Construct solutions to problems using student-created components, such as procedures, modules and/or objects.

07 Python While Loops

Topics:

- Loops

Standards

3B-AP-13: Illustrate the flow of execution of a recursive algorithm.

2-DA-07: Represent data using multiple encoding schemes.

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

2-AP-13: Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.

2-AP-17: Systematically test and refine programs using a range of test cases.

3A-AP-13: Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.

3A-AP-17: Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.

3A-AP-18: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.

3B-AP-11: Evaluate algorithms in terms of their efficiency, correctness, and clarity.

3B-AP-12: Compare and contrast fundamental data structures and their uses.

3B-AP-14: Construct solutions to problems using student-created components, such as procedures, modules and/or objects.

08 Python Functions

Topics:

- Functions

Standards

2-AP-14: Create procedures with parameters to organize code and make it easier to reuse.

2-DA-07: Represent data using multiple encoding schemes.

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

2-AP-13: Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.

2-AP-17: Systematically test and refine programs using a range of test cases.

3A-AP-13: Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.

3A-AP-17: Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.

3A-AP-18: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.

3B-AP-11: Evaluate algorithms in terms of their efficiency, correctness, and clarity.

3B-AP-12: Compare and contrast fundamental data structures and their uses.

3B-AP-14: Construct solutions to problems using student-created components, such as procedures, modules and/or objects.

09 Python Functions Parameters

Topics:

- Function parameters

Standards

2-AP-14: Create procedures with parameters to organize code and make it easier to reuse.

2-DA-07: Represent data using multiple encoding schemes.

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

2-AP-13: Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.

2-AP-17: Systematically test and refine programs using a range of test cases.

3A-AP-13: Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.

3A-AP-17: Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.

3A-AP-18: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.

3B-AP-11: Evaluate algorithms in terms of their efficiency, correctness, and clarity.

3B-AP-12: Compare and contrast fundamental data structures and their uses.

3B-AP-14: Construct solutions to problems using student-created components, such as procedures, modules and/or objects.

10 Multiple Parameters in Python Functions

Topics:

- Multiple Parameters in Python Functions

Standards

2-AP-14: Create procedures with parameters to organize code and make it easier to reuse.

2-DA-07: Represent data using multiple encoding schemes.

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

2-AP-13: Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.

2-AP-17: Systematically test and refine programs using a range of test cases.

3A-AP-13: Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.

3A-AP-17: Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.

3A-AP-18: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.

3B-AP-11: Evaluate algorithms in terms of their efficiency, correctness, and clarity.

3B-AP-12: Compare and contrast fundamental data structures and their uses.

3B-AP-14: Construct solutions to problems using student-created components, such as procedures, modules and/or objects.

11 Python Dictionaries

Topics:

- Python Dictionaries

Standards

2-DA-07: Represent data using multiple encoding schemes.

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

2-AP-13: Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.

2-AP-17: Systematically test and refine programs using a range of test cases.

3A-AP-13: Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.

3A-AP-17: Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.

3A-AP-18: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.

3B-AP-11: Evaluate algorithms in terms of their efficiency, correctness, and clarity.

3B-AP-12: Compare and contrast fundamental data structures and their uses.

3B-AP-14: Construct solutions to problems using student-created components, such as procedures, modules and/or objects.

12 Adding to and Removing from Dictionaries

Topics:

- Adding dictionaries

- Removing dictionaries

Standards

2-DA-07: Represent data using multiple encoding schemes.

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

2-AP-13: Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.

2-AP-17: Systematically test and refine programs using a range of test cases.

3A-AP-13: Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.

3A-AP-17: Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.

3A-AP-18: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.

3B-AP-11: Evaluate algorithms in terms of their efficiency, correctness, and clarity.

3B-AP-12: Compare and contrast fundamental data structures and their uses.

3B-AP-14: Construct solutions to problems using student-created components, such as procedures, modules and/or objects.

13 Dictionaries Continued

Topics:

- Dictionary Length
- Checking Dictionary Keys
- Converting from Lists to Dictionaries

Standards

2-DA-07: Represent data using multiple encoding schemes.

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

2-AP-13: Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.

2-AP-17: Systematically test and refine programs using a range of test cases.

3A-AP-13: Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.

3A-AP-17: Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.

3A-AP-18: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.

3B-AP-11: Evaluate algorithms in terms of their efficiency, correctness, and clarity.

3B-AP-12: Compare and contrast fundamental data structures and their uses.

3B-AP-14: Construct solutions to problems using student-created components, such as procedures, modules and/or objects.

14 Looping Through a Dictionary

Topics:

- Looping through a dictionary

Standards

3B-AP-13: Illustrate the flow of execution of a recursive algorithm.

2-DA-07: Represent data using multiple encoding schemes.

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

2-AP-13: Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.

2-AP-17: Systematically test and refine programs using a range of test cases.

3A-AP-13: Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.

3A-AP-17: Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.

3A-AP-18: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.

3B-AP-11: Evaluate algorithms in terms of their efficiency, correctness, and clarity.

3B-AP-12: Compare and contrast fundamental data structures and their uses.

3B-AP-14: Construct solutions to problems using student-created components, such

as procedures, modules and/or objects.

15 Looping through a Dictionary Continued

Topics:

- Printing Key Names with Loops in Dictionaries
- Printing Values with Loops in Dictionaries
- Printing both Key Names and Values with Loops in Dictionaries

Standards

2-DA-07: Represent data using multiple encoding schemes.

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

2-AP-13: Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.

2-AP-17: Systematically test and refine programs using a range of test cases.

3A-AP-13: Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.

3A-AP-17: Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.

3A-AP-18: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.

3B-AP-11: Evaluate algorithms in terms of their efficiency, correctness, and clarity.

3B-AP-12: Compare and contrast fundamental data structures and their uses.

3B-AP-14: Construct solutions to problems using student-created components, such as procedures, modules and/or objects.

16 Tuples

Topics:

- Tuples

Standards

2-DA-07: Represent data using multiple encoding schemes.

2-AP-11: Create clearly named variables that represent different data types and perform operations on their values.

2-AP-13: Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.

2-AP-17: Systematically test and refine programs using a range of test cases.

3A-AP-13: Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.

3A-AP-17: Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.

3A-AP-18: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.

3B-AP-11: Evaluate algorithms in terms of their efficiency, correctness, and clarity.

3B-AP-12: Compare and contrast fundamental data structures and their uses.

3B-AP-14: Construct solutions to problems using student-created components, such as procedures, modules and/or objects.

Cross Curriculum: Exploring Math through Python Code

Course Overview and Goals

This course is designed to help students explore math standards through a programming lens. This course is designed to help students in 6th - 8th grade and can be taught as a unit in their math classes. As they learn different programming concepts, they will also practice math principles aligned to the Common Core standards. There are no prerequisite courses for this unit. Students who have never programmed before can start with this mini unit. Students enjoy the variety of learning about programming as they learn math. There are no prerequisites for this course.

Who is it for?

The Exploring Math through Python Code course is designed for beginning Python students. No previous programming experience is required.

01 Variables in Python Math

Topics:

- Python Variables

Standards:

CCSS.MATH.CONTENT.6.EE.A.2: Write, read, and evaluate expressions in which letters stand for numbers.

CCSS.MATH.PRACTICE.MP1: Make sense of problems and persevere in solving them.

CCSS.MATH.CONTENT.7.EE.B.4: Use variables to represent quantities in a real-world or mathematical problem, and construct simple equations and inequalities to solve problems by reasoning about the quantities.

CCSS.MATH.PRACTICE.MP2: Reason abstractly and quantitatively.

CCSS.MATH.PRACTICE.MP5: Use appropriate tools strategically.

CCSS.MATH.PRACTICE.MP6: Attend to precision.

CCSS.MATH.PRACTICE.MP7: Look for and make use of structure.

02 Using Variables for Python Math

Topics:

- Python Variables

Standards:

CCSS.MATH.CONTENT.6.EE.A.2: Write, read, and evaluate expressions in which letters stand for numbers.

CCSS.MATH.CONTENT.6.EE.A.2.A: Write expressions that record operations with numbers and with letters standing for numbers. For example, express the calculation

"Subtract y from 5" as 5 - y.

CCSS.MATH.CONTENT.6.RP.A.1: Understand the concept of a ratio and use ratio language to describe a ratio relationship between two quantities.

CCSS.MATH.CONTENT.6.RP.A.3.B: Solve unit rate problems including those involving unit pricing and constant speed.

CCSS.MATH.CONTENT.7.EE.B.4: Use variables to represent quantities in a real-world or mathematical problem, and construct simple equations and inequalities to solve problems by reasoning about the quantities.

CCSS.MATH.PRACTICE.MP1: Make sense of problems and persevere in solving them.

CCSS.MATH.PRACTICE.MP2: Reason abstractly and quantitatively.

CCSS.MATH.PRACTICE.MP5: Use appropriate tools strategically.

CCSS.MATH.PRACTICE.MP6: Attend to precision.

CCSS.MATH.PRACTICE.MP7: Look for and make use of structure.

03 Inputs/Outputs Python Math

Topics:

- Inputs and outputs

Standards:

CCSS.MATH.CONTENT.6.EE.A.2: Write, read, and evaluate expressions in which letters stand for numbers.

CCSS.MATH.CONTENT.6.EE.A.2.A: Write expressions that record operations with numbers and with letters standing for numbers. For example, express the calculation "Subtract y from 5" as 5 - y.

CCSS.MATH.CONTENT.6.RP.A.1: Understand the concept of a ratio and use ratio language to describe a ratio relationship between two quantities.

CCSS.MATH.CONTENT.6.RP.A.3.B: Solve unit rate problems including those involving unit pricing and constant speed.

CCSS.MATH.CONTENT.7.EE.B.4: Use variables to represent quantities in a real-world or mathematical problem, and construct simple equations and inequalities to solve problems by reasoning about the quantities.

CCSS.MATH.PRACTICE.MP1: Make sense of problems and persevere in solving them.

CCSS.MATH.PRACTICE.MP2: Reason abstractly and quantitatively.

CCSS.MATH.PRACTICE.MP5: Use appropriate tools strategically.

CCSS.MATH.PRACTICE.MP6: Attend to precision.

CCSS.MATH.PRACTICE.MP7: Look for and make use of structure.

04 If Statements Python Math

Topics:

- If statements

Standards:

CCSS.MATH.CONTENT.6.RP.A.2: Understand the concept of a unit rate a/b associated with a ratio a:b with b ≠ 0, and use rate language in the context of a ratio relationship.

CCSS.MATH.CONTENT.6.RP.A.3.B: Solve unit rate problems including those involving unit pricing and constant speed.

CCSS.MATH.CONTENT.7.EE.B.4: Use variables to represent quantities in a real-world or mathematical problem, and construct simple equations and inequalities to solve problems by reasoning about the quantities.

CCSS.MATH.PRACTICE.MP1: Make sense of problems and persevere in solving them.

CCSS.MATH.PRACTICE.MP2: Reason abstractly and quantitatively.

CCSS.MATH.PRACTICE.MP5: Use appropriate tools strategically.

CCSS.MATH.PRACTICE.MP6: Attend to precision.

CCSS.MATH.PRACTICE.MP7: Look for and make use of structure.

05 Conditionals Python Math

Topics:

- Conditionals

Standards:

CCSS.MATH.CONTENT.6.NS.C.7.A: Interpret statements of inequality as statements about the relative position of two numbers on a number line diagram.

CCSS.MATH.CONTENT.6.NS.C.5: Understand that positive and negative numbers are used together to describe quantities having opposite directions or values

CCSS.MATH.CONTENT.7.EE.B.4: Use variables to represent quantities in a real-world or mathematical problem, and construct simple equations and inequalities to solve problems by reasoning about the quantities.

CCSS.MATH.CONTENT.6.NS.C.7.B: Write, interpret, and explain statements of order for rational numbers in real-world contexts.

CCSS.MATH.CONTENT.6.NS.C.7.C: Understand the absolute value of a rational number as its distance from 0 on the number line

CCSS.MATH.CONTENT.6.NS.C.7.D: Distinguish comparisons of absolute value from statements about order.

CCSS.MATH.CONTENT.6.EE.A.2: Write, read, and evaluate expressions in which letters stand for numbers.

CCSS.MATH.CONTENT.6.EE.A.2.A: Write expressions that record operations with numbers and with letters standing for numbers.

CCSS.MATH.CONTENT.7.NS.A.1.A: Describe situations in which opposite quantities combine to make 0.

CCSS.MATH.PRACTICE.MP1: Make sense of problems and persevere in solving them.

CCSS.MATH.CONTENT.7.RP.A.2.C: Represent proportional relationships by equations.

CCSS.MATH.CONTENT.7.RP.A.3: Use proportional relationships to solve multistep ratio and percent problems.

CCSS.MATH.PRACTICE.MP2: Reason abstractly and quantitatively.

CCSS.MATH.PRACTICE.MP5: Use appropriate tools strategically.

CCSS.MATH.PRACTICE.MP6: Attend to precision.

CCSS.MATH.PRACTICE.MP7: Look for and make use of structure.

06 Python Geometry

Topics:

- Geometry

Standards:

CCSS.MATH.CONTENT.6.EE.A.2.C: Evaluate expressions at specific values of their variables.

CCSS.MATH.CONTENT.7.G.A.3: Describe the two-dimensional figures that result from slicing three-dimensional figures, as in plane sections of right rectangular prisms and right rectangular pyramids.

CCSS.MATH.CONTENT.7.G.B.4: Know the formulas for the area and circumference of a circle and use them to solve problems

CCSS.MATH.CONTENT.7.G.B.6: Solve real-world and mathematical problems involving area, volume and surface area of two- and three-dimensional objects composed of triangles, quadrilaterals, polygons, cubes, and right prisms.

CCSS.MATH.CONTENT.8.G.C.9: Know the formulas for the volumes of cones, cylinders, and spheres and use them to solve real-world and mathematical problems.

CCSS.MATH.PRACTICE.MP1: Make sense of problems and persevere in solving them.

CCSS.MATH.PRACTICE.MP2: Reason abstractly and quantitatively.

CCSS.MATH.PRACTICE.MP5: Use appropriate tools strategically.

CCSS.MATH.PRACTICE.MP6: Attend to precision.

CCSS.MATH.PRACTICE.MP7: Look for and make use of structure.

Skill Struck ®

Cross Curriculum: Exploring Conservation with Python Code

Course Overview and Goals

This course explores conservation principles by following the movement of an elephant seal over the course of a few months. Using Python code, students build the code for the instrument that would work to document important data points. Together, they generate ideas about the animal's movements and what can be done to protect and preserve the species. There are no prerequisites for this course.

Who is it for?

The Exploring Conservation with Python Code course is designed for beginning Python students. No previous programming experience is required.

01 Animal Tracking and Variables

Topics:

- Variables

Standards:

MS-ESS3-4 Earth and Human Activity: Construct an argument supported by evidence for how increases in human population and per-capita consumption of natural resources impact Earth's systems.

MS-ESS3-5 Earth and Human Activity: Ask questions to clarify evidence of the factors that have caused the rise in global temperatures over the past century.

MS-LS2-1 Ecosystems: Interactions, Energy, and Dynamics: Analyze and interpret data to provide evidence for the effects of resource availability on organisms and populations of organisms in an ecosystem.

MS-LS2-2 Ecosystems: Interactions, Energy, and Dynamics: Construct an explanation that predicts patterns of interactions among organisms across multiple ecosystems.

MS-LS2-3 Ecosystems: Interactions, Energy, and Dynamics: Develop a model to describe the cycling of matter and flow of energy among living and nonliving parts of an ecosystem.

MS-LS2-4 Ecosystems: Interactions, Energy, and Dynamics: Construct an argument supported by empirical evidence that changes to physical or biological components of an ecosystem affect populations.

02 Animal Tracking and Variable Types

Topics:

- Variable Types

Standards:

MS-ESS3-4 Earth and Human Activity: Construct an argument supported by evidence for how increases in human population and per-capita consumption of natural resources impact Earth's systems.

MS-ESS3-5 Earth and Human Activity: Ask questions to clarify evidence of the factors that have caused the rise in global temperatures over the past century.

MS-LS2-1 Ecosystems: Interactions, Energy, and Dynamics: Analyze and interpret data to provide evidence for the effects of resource availability on organisms and populations of organisms in an ecosystem.

MS-LS2-2 Ecosystems: Interactions, Energy, and Dynamics: Construct an explanation that predicts patterns of interactions among organisms across multiple ecosystems.

MS-LS2-3 Ecosystems: Interactions, Energy, and Dynamics: Develop a model to describe the cycling of matter and flow of energy among living and nonliving parts of an ecosystem.

MS-LS2-4 Ecosystems: Interactions, Energy, and Dynamics: Construct an argument supported by empirical evidence that changes to physical or biological components of an ecosystem affect populations.

03 Animal Tracking and Inputs/Outputs

Topics:

- Inputs and outputs

Standards:

MS-ESS3-4 Earth and Human Activity: Construct an argument supported by evidence for how increases in human population and per-capita consumption of natural resources impact Earth's systems.

MS-ESS3-5 Earth and Human Activity: Ask questions to clarify evidence of the factors that have caused the rise in global temperatures over the past century.

MS-LS2-1 Ecosystems: Interactions, Energy, and Dynamics: Analyze and interpret data to provide evidence for the effects of resource availability on organisms and populations of organisms in an ecosystem.

MS-LS2-2 Ecosystems: Interactions, Energy, and Dynamics: Construct an explanation that predicts patterns of interactions among organisms across multiple ecosystems.

MS-LS2-3 Ecosystems: Interactions, Energy, and Dynamics: Develop a model to describe the cycling of matter and flow of energy among living and nonliving parts of an ecosystem.

MS-LS2-4 Ecosystems: Interactions, Energy, and Dynamics: Construct an argument supported by empirical evidence that changes to physical or biological components of an ecosystem affect populations.

04 Animal Tracking and Converting Data Types

Topics:

- Converting data types

Standards:

MS-ESS3-2 Earth and Human Activity: Analyze and interpret data on natural hazards to forecast future catastrophic events and inform the development of technologies to mitigate their effects.

MS-ESS3-3 Earth and Human Activity: Apply scientific principles to design a method for monitoring and minimizing a human impact on the environment.*

MS-ESS3-4 Earth and Human Activity: Construct an argument supported by evidence for how increases in human population and per-capita consumption of natural resources impact Earth's systems.

MS-LS2-1 Ecosystems: Interactions, Energy, and Dynamics: Analyze and interpret data to provide evidence for the effects of resource availability on organisms and populations of organisms in an ecosystem.

MS-LS2-2 Ecosystems: Interactions, Energy, and Dynamics: Construct an explanation that predicts patterns of interactions among organisms across multiple ecosystems.

MS-LS2-3 Ecosystems: Interactions, Energy, and Dynamics: Develop a model to describe the cycling of matter and flow of energy among living and nonliving parts of an ecosystem.

MS-LS2-4 Ecosystems: Interactions, Energy, and Dynamics: Construct an argument supported by empirical evidence that changes to physical or biological components of an ecosystem affect populations.

Skill Struck ®

05 Animal Tracking and If Statements

Topics:

- If statements

Standards:

MS-ESS3-2 Earth and Human Activity: Analyze and interpret data on natural hazards to forecast future catastrophic events and inform the development of technologies to mitigate their effects.

MS-ESS3-3 Earth and Human Activity: Apply scientific principles to design a method for monitoring and minimizing a human impact on the environment.*

MS-ESS3-4 Earth and Human Activity: Construct an argument supported by evidence for how increases in human population and per-capita consumption of natural resources impact Earth's systems.

MS-LS2-1 Ecosystems: Interactions, Energy, and Dynamics: Analyze and interpret data to provide evidence for the effects of resource availability on organisms and populations of organisms in an ecosystem.

MS-LS2-2 Ecosystems: Interactions, Energy, and Dynamics: Construct an explanation that predicts patterns of interactions among organisms across multiple ecosystems.

MS-LS2-3 Ecosystems: Interactions, Energy, and Dynamics: Develop a model to describe the cycling of matter and flow of energy among living and nonliving parts of an ecosystem.

MS-LS2-4 Ecosystems: Interactions, Energy, and Dynamics: Construct an argument supported by empirical evidence that changes to physical or biological components of an ecosystem affect populations.

06 Animal Tracking with Conditionals

Topics:

- Conditionals

Standards:

MS-ESS3-2 Earth and Human Activity: Analyze and interpret data on natural hazards to forecast future catastrophic events and inform the development of technologies to mitigate their effects.

MS-ESS3-3 Earth and Human Activity: Apply scientific principles to design a method for monitoring and minimizing a human impact on the environment.*

MS-ESS3-4 Earth and Human Activity: Construct an argument supported by evidence for how increases in human population and per-capita consumption of natural resources impact Earth's systems.

MS-LS2-1 Ecosystems: Interactions, Energy, and Dynamics: Analyze and interpret data to provide evidence for the effects of resource availability on organisms and populations of organisms in an ecosystem.

MS-LS2-2 Ecosystems: Interactions, Energy, and Dynamics: Construct an explanation that predicts patterns of interactions among organisms across multiple ecosystems.

MS-LS2-3 Ecosystems: Interactions, Energy, and Dynamics: Develop a model to describe the cycling of matter and flow of energy among living and nonliving parts of an ecosystem.

MS-LS2-4 Ecosystems: Interactions, Energy, and Dynamics: Construct an argument supported by empirical evidence that changes to physical or biological components of an ecosystem affect populations.

07 Animal Tracking Engineering Project

Topics:

- Variables
- Converting data types
- If statements
- Conditionals
- Inputs/Outputs

Standards:

MS-ETS1-1 Engineering Design: Define the criteria and constraints of a design problem with sufficient precision to ensure a successful solution, taking into account relevant scientific principles and potential impacts on people and the natural environment that may limit possible solutions.

MS-ETS1-2 Engineering Design: Evaluate competing design solutions using a systematic process to determine how well they meet the criteria and constraints of the problem.

MS-ETS1-3 Engineering Design: Analyze data from tests to determine similarities and differences among several design solutions to identify the best characteristics of each that can be combined into a new solution to better meet the criteria for success.

MS-ETS1-4 Engineering Design: Develop a model to generate data for iterative testing and modification of a proposed object, tool, or process such that an optimal design can be achieved.