Unit 2 - Coding - CS Express Copied from: Introduction to Coding: Computer Programming, Copied on: 02/21/22

Content Area: **Sample Content Area**
Course(s): **Sample Course**
Time Period: **SeptOct**
Length: **Sample Length & Grade Level**
Status: **Published**

Coding in CS Express

Department of Curriculum and Instruction

Belleville Public Schools

Curriculum Guide

Introduction to Coding, Grades 9-12

Coding in CS Express

Belleville Board of Education

102 Passaic Avenue

Belleville, NJ 07109

Prepared by: Teacher, Corey Woodring

Dr. Richard Tomko, Ph.D., M.J., Superintendent of Schools

Ms. LucyAnn Demikoff, Director of Curriculum and Instruction K-12

Ms. Nicole Shanklin, Director of Elementary Education

Mr. Joseph Lepo, Director of Secondary Education

Board Approved:

Unit Overview

This unit provide a fun, engaging way for students to learn basic computer science concepts.
Students learn programming concepts, computational thinking, and develop problem-solving skills and persistence.
Students will create computer programs including interactive games and creative projects they can share.

This unit provides an overview of the main applications of programming: computer science, web development, and data science.
It also teaches important concepts that you'll find in every programming language, such as variables, functions, and control flow. After completing this unit, students will understand key programming terms and be familiar with key programming concepts.

Enduring Understanding

Creative development can be an essential process for creating computational artifacts.
Computing enables people to use creative development processes to create computational artifacts for creative expression or to solve a problem.
Computing can extend traditional forms of human expression and experience.
Multiple levels of abstraction are used to write programs or create other computational artifacts
Algorithms are precise sequences of instructions for processes that can be executed by a computer and are implemented using programming languages.
Programs can be developed for creative expression, to satisfy personal curiosity, to create new knowledge, or to solve problems (to help people, organizations, or society).
People write programs to execute algorithms.
Programming is facilitated by appropriate abstractions.
Programs are developed, maintained, and used by people for different purposes.
Programming uses mathematical and logical concepts.
Computing enhances communication, interaction, and cognition.

Essential Questions

Can I Write Code?

HOW DO YOU DEFINE 'CODING,' AND WHY IS IT IMPORTANT?

What is the difference between an iterative and a recursive algorithm and how should each be used?

Why are comments, indentation, and naming conventions important to the readability of a program?

Why is it important to know the data type of a variable?

Exit Skills

Students will be able too:
Use a variety of media to develop and deepen understanding of a topic or idea.
Develop a model or prototype for iterative testing and refinement.
Apply the principles of writing code
Students will be able to:

Download and explore the IDLE IDE.
Utilize the IDE to write simple programs.
Use pseudocode, flowcharts, or words to describe the process for changing a tire or cooking a turkey.
Describe the procedures and subprocedures needed to perform complicated tasks.
Understand the top-down execution of basic coding scripts.
Write simple programs utilizing correct style and naming conventions.

New Jersey Student Learning Standards (NJSLS-S)

Upon completion of this section, please remove all remaining descriptions, notes, outlines, examples and/or illustrations that are not needed or used.

Please list only the content-level and cross-curricular **New Jersey Student Learning Standards** applicable to the unit. **Do not list standards that are not used in the unit.**

	A computing system involves interaction among the user, hardware, application software, and system software.
8.1.12.NI.1	Evaluate the scalability and reliability of networks, by describing the relationship between routers, switches, servers, topology, and addressing.
8.1.12.AP.1	Design algorithms to solve computational problems using a combination of original and existing algorithms.

Interdisciplinary Connections

Upon completion of this section, please remove all remaining descriptions, notes, outlines, examples and/or illustrations that are not

needed or used.

Please list all and any additional **Interdisciplinary Connections/Cross-Curricular** New Jersey Student Learning Standards that link to this unit, and which are not included in the NJSLS section above.

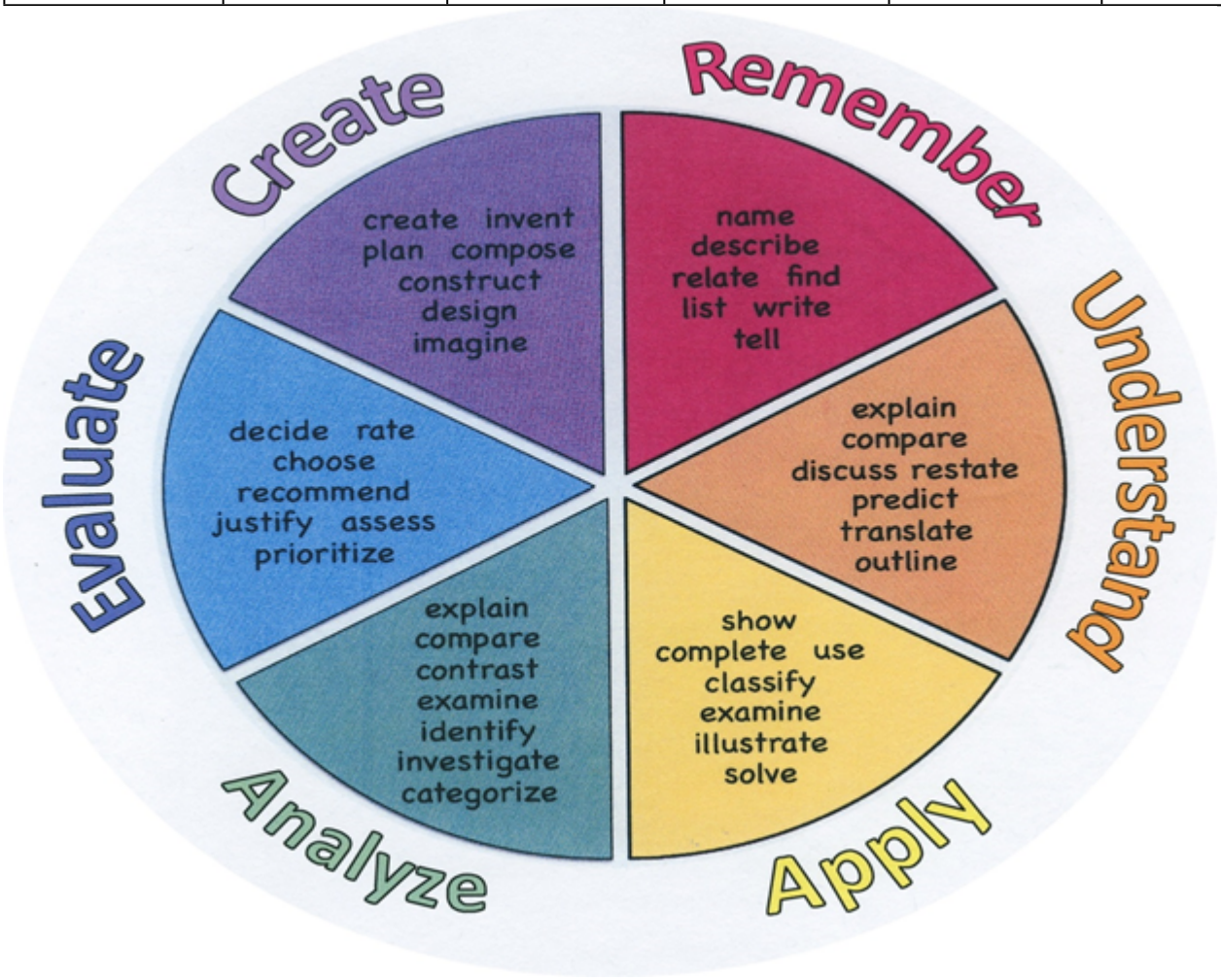
SCI.HS-ETS1-4	Use a computer simulation to model the impact of proposed solutions to a complex real-world problem with numerous criteria and constraints on interactions within and between systems relevant to the problem.
SCI.HS-ETS1-3	Evaluate a solution to a complex real-world problem based on prioritized criteria and trade-offs that account for a range of constraints, including cost, safety, reliability, and aesthetics, as well as possible social, cultural, and environmental impacts.
SCI.HS-ETS1-2	Design a solution to a complex real-world problem by breaking it down into smaller, more manageable problems that can be solved through engineering.

Learning Objectives

SOLVING PROBLEMS WITH CODE:
Can I Write Code? Learning Objectives: I can:
- Use a variety of media to develop and deepen understanding of a topic or idea.
- Develop a model or prototype for iterative testing and refinement.
- Apply the principles of writing code

Remember	Understand	Apply	Analyze	Evaluate	Create
Choose	Classify	Choose	Categorize	Appraise	Combine
Describe	Defend	Dramatize	Classify	Judge	Compose
Define	Demonstrate	Explain	Compare	Criticize	Construct
Label	Distinguish	Generalize	Differentiate	Defend	Design
List	Explain	Judge	Distinguish	Compare	Develop
Locate	Express	Organize	Identify	Assess	Formulate
Match	Extend	Paint	Infer	Conclude	Hypothesize
Memorize	Give Examples	Prepare	Point out	Contrast	Invent
Name	Illustrate	Produce	Select	Critique	Make
Omit	Indicate	Select	Subdivide	Determine	Originate
Recite	Interrelate	Show	Survey	Grade	Organize
Select	Interpret	Sketch	Arrange	Justify	Plan
State	Infer	Solve	Breakdown	Measure	Produce
Count	Match	Use	Combine	Rank	Role Play
Draw	Paraphrase	Add	Detect	Rate	Drive
Outline	Represent	Calculate	Diagram	Support	Devise
Point	Restate	Change	Discriminate	Test	Generate
Quote	Rewrite	Classify	Illustrate		Integrate
Recall	Select	Complete	Outline		Prescribe
Recognize	Show	Compute	Point out		Propose
Repeat	Summarize	Discover	Separate		Reconstruct
Reproduce	Tell	Divide			Revise
	Translate	Examine			Rewrite
	Associate	Graph			Transform
	Compute	Interpolate			
	Convert	Manipulate			

	Discuss Estimate Extrapolate Generalize Predict	Modify Operate Subtract			



Suggested Activities & Best Practices

A Royal Battle with Events

In this **mini-project**, students will have the opportunity to learn how to use events in Play Lab and apply all of the coding skills that they've learned to create an animated game.

On the Move with Events

In this **mini-project**, students will have the opportunity to learn how to use events in Play Lab and to apply all of the coding skills they've learned to create an animated game. It's time to get creative and make a story in the Play Lab!

End of Unit Project

This **project** lesson takes students through the process of designing, developing, and showcasing new projects!

Code.org

Assessment Evidence - Checking for Understanding (CFU)

Quizzes

Unit tests-summative assessment

Admit/Exit tickets-formative assessment

Web Based Assessments -alternate assessment

Create a Multimedia poster-benchmark assessment

- Admit Tickets
- Anticipation Guide
- Common Benchmarks
- Compare & Contrast
- Create a Multimedia Poster
- DBQ's
- Define
- Describe
- Evaluate
- Evaluation rubrics
- Exit Tickets
- Explaining
- Fist- to-Five or Thumb-Ometer
- Illustration
- Journals
- KWL Chart
- Learning Center Activities

- Multimedia Reports
- Newspaper Headline
- Outline
- Question Stems
- Quickwrite
- Quizzes
- Red Light, Green Light
- Self- assessments
- Socratic Seminar
- Study Guide
- Surveys
- Teacher Observation Checklist
- Think, Pair, Share
- Think, Write, Pair, Share
- Top 10 List
- Unit review/Test prep
- Unit tests
- Web-Based Assessments
- Written Reports

Primary Resources & Materials

Code Combat Platform, **Code.org** Platform, Game Salad Platform

Ancillary Resources

1. LightBot
2. SpriteBox
3. Hour of Code
4. Code.org
5. Scratch
6. Grasshopper App
7. CS First

.

Technology Infusion

You Tube

Khan Academy

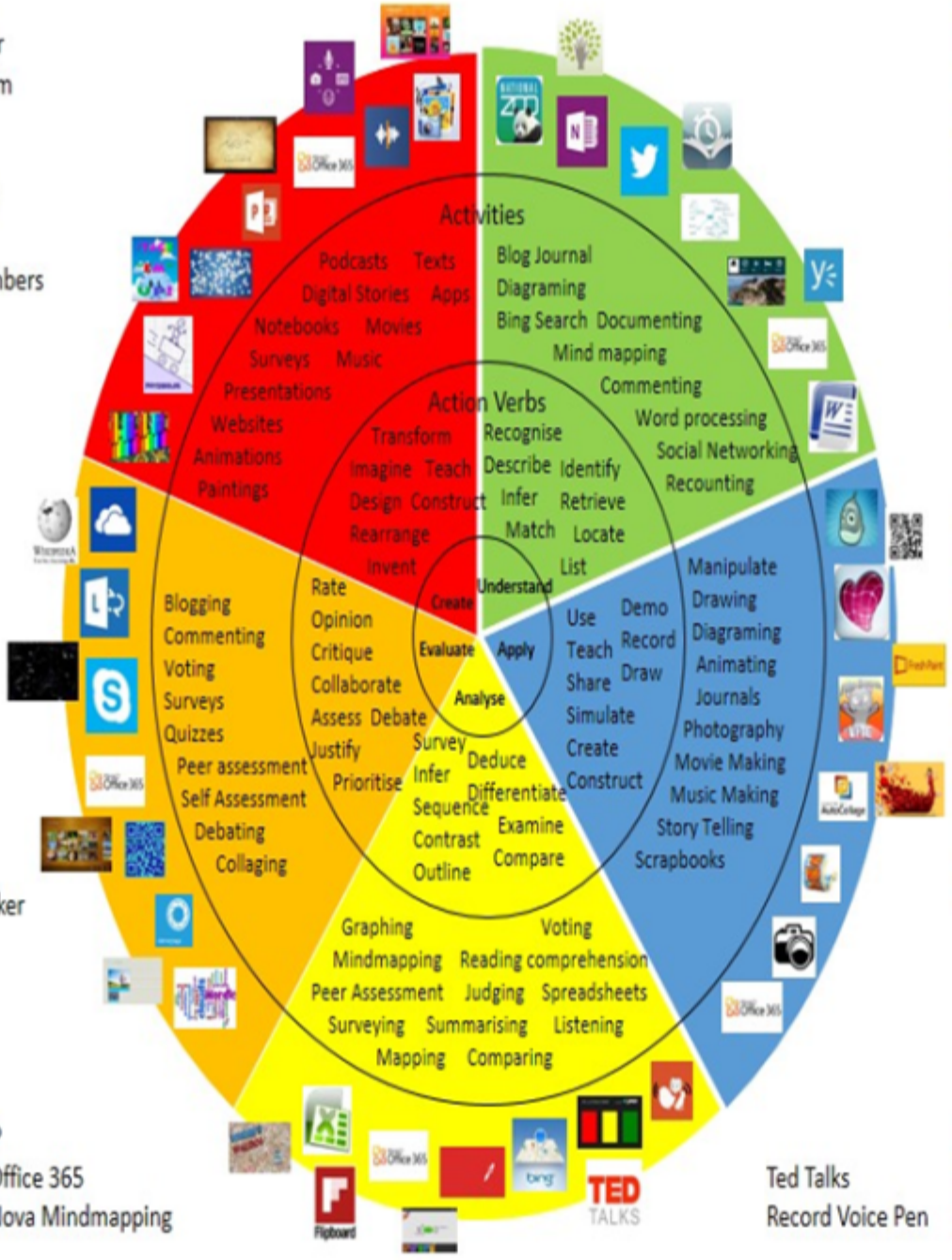
Office 365

Technology Infusion and/or strategies are integrated into this unit to enhance learning

Win 8.1 Apps/Tools Pedagogy Wheel

Podcasts
Photostory 3
Kid Story Builder
Music Maker Jam
Paint A Story
Office 365
MS PowerPoint
Stack 'Em Up
NqSquared Numbers
Physamajig
Xylophone 8

Wikipedia
Skydrive
Lync
SkyMap
Skype
Office 365
Puzzle Touch
Easy QR
Memorylage
Life Moments
Word Cloud Maker



Where's Waldo?
MS Excel Office 365
Flipboard Nova Mindmapping

Ted Talks
Record Voice Pen

Originally taken from http://www.coetail.com/vzimmer/files/2013/02/iPadagogy-Wheel.001.jpg
And adapted for Windows 8.1 devices by Charlotte Beckhurst @CharBeckhurst

Alignment to 21st Century Skills & Technology

Mastery and infusion of **21st Century Skills & Technology** and their Alignment to the core content areas is essential to student learning. The core content areas include:

- English Language Arts;
- Mathematics;
- Science and Scientific Inquiry (Next Generation);
- Social Studies, including American History, World History, Geography, Government and Civics, and Economics;
- World languages;
- Technology;
- Visual and Performing Arts.

SCI.HS	Engineering Design
SCI.HS-ETS1-1	Analyze a major global challenge to specify qualitative and quantitative criteria and constraints for solutions that account for societal needs and wants.
SCI.HS-ETS1-3	Evaluate a solution to a complex real-world problem based on prioritized criteria and trade-offs that account for a range of constraints, including cost, safety, reliability, and aesthetics, as well as possible social, cultural, and environmental impacts.
SCI.HS-ETS1-4	Use a computer simulation to model the impact of proposed solutions to a complex real-world problem with numerous criteria and constraints on interactions within and between systems relevant to the problem.
SCI.HS-ETS1-2	Design a solution to a complex real-world problem by breaking it down into smaller, more manageable problems that can be solved through engineering.

21st Century Skills/Interdisciplinary Themes

Upon completion of this section, please remove all remaining descriptions, notes, outlines, examples and/or illustrations that are not needed or used.

Please list only the **21st Century/Interdisciplinary Themes** that will be incorporated into this unit.

- Communication and Collaboration
- Creativity and Innovation
- Critical thinking and Problem Solving
- ICT (Information, Communications and Technology) Literacy
- Information Literacy
- Life and Career Skills
- Media Literacy

21st Century Skills

Upon completion of this section, please remove all remaining descriptions, notes, outlines, examples and/or illustrations that are not

needed or used.

Please list only the **21st Century Skills** that will be incorporated into this unit.

- Civic Literacy
- Environmental Literacy
- Financial, Economic, Business and Entrepreneurial Literacy
- Global Awareness
- Health Literacy

Differentiation

- Exemplars
- Show students how to find the hints, methods reference cards, error messages, and sample code provided within each level.
- Students struggling with a given level will be automatically directed to additional practice levels within the game.

- Meeting with small groups to re-teach an idea or skill for struggling learners, or to extend the thinking or skills of advanced learners.

  . Varying the length of time a student may take to complete a task in order to provide additional support for a struggling learner or to encourage an advanced learner to pursue a topic in greater depth.

  Using rubrics that match and extend students' varied skills levels

Differentiations:

- Small group instruction
- Small group assignments
- Extra time to complete assignments
- Pairing oral instruction with visuals
- Repeat directions
- Use manipulatives
- Center-based instruction
- Token economy
- Study guides
- Teacher reads assessments allowed
- Scheduled breaks
- Rephrase written directions
- Multisensory approaches
- Additional time
- Preview vocabulary

- Preview content & concepts
- Story guides
- Behavior management plan
- Highlight text
- Student(s) work with assigned partner
- Visual presentation
- Assistive technology
- Auditory presentations
- Large print edition
- Dictation to scribe
- Small group setting

Hi-Prep Differentiations:

- Alternative formative and summative assessments
- Choice boards
- Games and tournaments
- Group investigations
- Guided Reading
- Independent research and projects
- Interest groups
- Learning contracts
- Leveled rubrics
- Literature circles
- Multiple intelligence options
- Multiple texts
- Personal agendas
- Project-based learning
- Problem-based learning
- Stations/centers
- Think-Tac-Toes
- Tiered activities/assignments
- Tiered products
- Varying organizers for instructions

Lo-Prep Differentiations

- Choice of books or activities
- Cubing activities
- Exploration by interest
- Flexible grouping
- Goal setting with students
- Jigsaw
- Mini workshops to re-teach or extend skills
- Open-ended activities
- Think-Pair-Share
- Reading buddies
- Varied journal prompts
- Varied supplemental materials

Special Education Learning (IEP's & 504's)

- **Exemplars**
- If students struggle with breaking down problems, you can use the printable Engineering Cycle Worksheet to reinforce a step-by-step problem-solving approach.
- If students struggle to follow correct syntax, provide a copy of the printable Python Syntax Guide or JavaScript Syntax Guide

IEP:

1. Adherence to the students' Individualized Learning Plan.

2. Students will have extra time or fewer assignments, one-to-one assistance, and group work will often be enlisted.

3. Students may use speech-to-text or audio/video record assignments

4. Teacher may adapt learning style to fit the needs of the child.

5. Teacher will use graphic organizer to visually help students plan out their work.

6. The teacher will scaffold the lesson with a slow release from assisted support with guided practice to independent practice.

7. Front-loaded notes to enable students to more accurately follow along with teacher's instruction.

8. Step-by-step directions written out for students.

504:

1. Any necessary accommodations will be made as outlined in students' 504 plan.

2. Preferential seating while teacher is lecturing, explaining, etc.

3. Extended time on projects or assessments.

4. Verbal, visual or technology aids.

5. Use of digital audio-visual materials, such as YouTube, to assist the child with directions.

6. Students will have access to "How To" videos, so they may more easily follow along with directions for their assignments.

- printed copy of board work/notes provided
- additional time for skill mastery
- assistive technology
- behavior management plan
- Center-Based Instruction
- check work frequently for understanding
- computer or electronic device utilizes
- extended time on tests/ quizzes
- have student repeat directions to check for understanding
- highlighted text visual presentation
- modified assignment format
- modified test content
- modified test format
- modified test length
- multi-sensory presentation
- multiple test sessions
- preferential seating
- preview of content, concepts, and vocabulary
- Provide modifications as dictated in the student's IEP/504 plan
- reduced/shortened reading assignments
- Reduced/shortened written assignments
- secure attention before giving instruction/directions
- shortened assignments
- student working with an assigned partner
- teacher initiated weekly assignment sheet
- Use open book, study guides, test prototypes

English Language Learning (ELL)

Exemplars: English Language Learners: 1. Most of the games have instructions available in other languages, and Google translate will be used as necessary. Students will be allowed to work with partners. 2. Sheltered Instruction Observation Protocol (SIOP) – instructional model that helps teachers plan and deliver lessons that allow English learners the ability to acquire academic knowledge as they develop English language proficiency. 3. Provide leveled texts or translations by using Google Chrome Extension: Snap&Read, which is a reading tool that can cover the most diverse reading needs. It features Read Aloud, Dynamic Text Leveling, Translation, and Study tools, such as pulling text into an outline, then organizing it and adding notes. 4. Teacher creates a culturally responsive classroom. 5. Teacher employs teaching strategies and learning resources that make content comprehensible. 6. Teacher employs Total Physical Response (TPR) – uses a direct action to help students internalize new language

Using videos, illustrations, pictures, and drawings to explain or clarify

English Language Learning adaptations that will be employed in the unit, using the ones identified below.

- teaching key aspects of a topic. Eliminate nonessential information
- using videos, illustrations, pictures, and drawings to explain or clarif
- allowing products (projects, timelines, demonstrations, models, drawings, dioramas, poster boards, charts, graphs, slide shows, videos, etc.) to demonstrate student's learning;
- allowing students to correct errors (looking for understanding)
- allowing the use of note cards or open-book during testing
- decreasing the amount of workpresented or required
- having peers take notes or providing a copy of the teacher's notes
- modifying tests to reflect selected objectives
- providing study guides
- reducing or omitting lengthy outside reading assignments
- reducing the number of answer choices on a multiple choice test
- tutoring by peers
- using computer word processing spell check and grammar check features
- using true/false, matching, or fill in the blank tests in lieu of essay tests

At Risk

Please identify Intervention Strategies that will be employed in the unit, using the ones identified below.

- allowing students to correct errors (looking for understanding)
- teaching key aspects of a topic. Eliminate nonessential information
- allowing products (projects, timelines, demonstrations, models, drawings, dioramas, poster boards, charts, graphs, slide shows, videos, etc.) to demonstrate student's learning
- allowing students to select from given choices
- allowing the use of note cards or open-book during testing
- collaborating (general education teacher and specialist) to modify vocabulary, omit or modify items to reflect objectives for the student, eliminate sections of the test, and determine how the grade will be determined prior to giving the test.
- decreasing the amount of workpresented or required
- having peers take notes or providing a copy of the teacher's notes
- marking students' correct and acceptable work, not the mistakes
- modifying tests to reflect selected objectives
- providing study guides
- reducing or omitting lengthy outside reading assignments
- reducing the number of answer choices on a multiple choice test

- tutoring by peers
- using authentic assessments with real-life problem-solving
- using true/false, matching, or fill in the blank tests in lieu of essay tests
- using videos, illustrations, pictures, and drawings to explain or clarify

Talented and Gifted Learning (T&G)

Exemplars:

Gifted and Talented:

1. Students will have the opportunity to publish their writing (reviews/analysis) online, submit their projects to developers, and enter in game design competitions. They may also extend their investigation to some video games, possibly extending their final projects into another format.

2. Teacher can use a pre-assessment to determine students' knowledge of standard being taught in lesson and then provide an extension activity for students

3. Compact lesson

4. Provide students with problem-based learning activity using multiple standards from the unit.

Higher order, critical & creative thinking skills, and discovery

Talented and Gifted adaptations that will be employed in the unit, using the ones identified below.

- Above grade level placement option for qualified students
- Advanced problem-solving
- Allow students to work at a faster pace
- Cluster grouping
- Complete activities aligned with above grade level text using Benchmark results
- Create a blog or social media page about their unit
- Create a plan to solve an issue presented in the class or in a text
- Debate issues with research to support arguments
- Flexible skill grouping within a class or across grade level for rigor
- Higher order, critical & creative thinking skills, and discovery
- Multi-disciplinary unit and/or project
- Teacher-selected instructional strategies that are focused to provide challenge, engagement, and growth opportunities
- Utilize exploratory connections to higher-grade concepts
- Utilize project-based learning for greater depth of knowledge

Sample Lesson

https://docs.google.com/document/d/1HUE7g2tqk-_3wiOU4cEFfE4CoOQ5kMzZQc-BTXdco5E/edit?usp=sharing

Lesson 1 - Basic Syntax

(Levels 1-8)

Summary

These levels introduce basic concepts and vocabulary, including syntax, strings, and arguments. Students find that sequencing is critical to coding because when a computer runs a program, it executes every command in the order it is given, from start to finish. Students are also introduced to commenting code, a common practice used by programmers to document and communicate about their work.

The opening classroom activity introduces Python syntax and the importance of order in a sequence of instructions, or algorithm. As the teacher, you will imitate a robot that executes the commands given by the class. By the end of the activity, the class will collaboratively write a program something like this:

```
teacher.pickUpBall()
teacher.turnRight()
teacher.moveForward()
teacher.moveForward()
teacher.turnLeft()
teacher.moveForward()
teacher.dropBall()
```

Materials

- Desk or table
- Recycling bin
- Balls of paper to recycle
- Optional: Progress Journal
- Optional: Engineering Cycle Worksheet
- Optional: Python Syntax Guide or JavaScript Syntax Guide

Learning Objectives

- Use correct syntax when writing code.
- Use proper sequencing when writing code.
- Use arguments to input information into a method.
- Use strings to input text data.

- Use comments to document code.
- Understand and use basic vocabulary: algorithm, argument, code, method, program, sequence, syntax, string

Standards

- **CSTA: 1A-AP-10** Develop programs with sequences and simple loops, to express ideas or address a problem.
- **CCSS-Math: MP.1** Make sense of problems and persevere in solving them.
- **CCSS-Math: MP.6** Attend to precision.

Opening Activity (15 minutes): *Recycling Robot*

Explain

Explain the following terms to students:

- **Syntax** is how we write code. Just as spelling and grammar are important in writing narratives and essays, syntax is important when writing code. Humans are good at figuring out what something means, even if it isn't exactly correct, but computers aren't that smart, and they need you to write very precisely.
- **Sequence** is the order of the commands in a program. Computers follow commands in exactly the order they are written.
- **Objects** are the building blocks of Python. They are things or characters that can perform actions. In the game, your hero is an object. The actions an object performs are called **methods**. For example, `moveRight()` is a method. Method names are always followed by parentheses.

Write the sample code `hero.moveRight()` on the board, and describe the components:

- This is read aloud as "hero dot move right", where `hero` is the object, and `moveRight` is the method.
- The period, capitalization, and parentheses are essential parts of the syntax that must be exactly right:
    - Period: separates the object from the method
    - Capital letters: used to show the start of a new word when a period or space can't be used (this is called "camel case")
    - Parentheses: create a place where a programmer could add extra details, or **arguments** to a method.

Interact

At the front of the class, set some scrunched up paper balls on a flat surface. Place the recycling bin a few steps away. Explain that you are a recycling robot, and the class's job is to program you.

The robot is a Python object. Choose a name for yourself and write it on the board, beginning with a lowercase letter. For example:

```
teacher
```

To make the robot perform an action, students have to call a method. Write a dot after your object name, then have the class decide what the first action should be. After the dot, write the method name using camel case followed by empty parentheses. For example:

```
teacher.pickUpBall()
```

Off to one side, draw a “Run” button and have a volunteer press it. As a robot, execute the program *precisely* as the students have written it.

Invite students to work together to add commands to the program one at a time until you can successfully get a ball into the recycling bin. They can test the program at any time by pressing the "Run" button. Each time they press "Run", you should execute every command from start to finish exactly as written. If there is an error in the syntax, make a funny beeping sound and stop. After each test, reset yourself and have the class revise, or *debug*, the program until it works.

Discuss

Use one or more of the following discussion questions to prompt a brief reflection:

Why are sequence and syntax important?

Sample Response:

> Computers only do exactly what you tell them, so the sequence is important because if the computer does the steps in the wrong order, the program doesn't turn out right. If there is an error in the syntax, the computer doesn't know how to read it at all.

How is the way a computer reads instructions different from the way a human would understand them?

Sample Response:

> Humans can use their own knowledge and other clues to figure things out if they don't make sense. Computers can only execute exactly what they are told.

Coding Time (30-40 minutes)

Tell students they will be playing Levels 1 - 8 today. Allow students to move through these levels at their own pace. Circulate and assist, calling attention to the Hints button in the top right corner of each level as needed.

We recommend stopping students after Level 8 and using the next lesson plan to introduce the next concepts before beginning Level 9.

Look Out For:

- Initially, some students may want to type and run one command at a time. Explain to them that the code must contain all the instructions for the program from start to finish, like a story: it has a beginning, a middle, and an end. Every time you click Start, the hero returns to the beginning of the level, and the full program runs again.

Closure (5 minutes)

Use one or more of the following questions to prompt reflection. You can facilitate a short discussion, or have students submit written responses on Exit Tickets.

Explain how to play CodeCombat to someone who has never played before. Use as many programming

terms as you can.

Sample Response:

> You have to move to the gem without hitting the spikes by writing a program. I learned that you have to type the object name first, like "hero." then the method to make them do an action. You have to spell it right and put () at the end. You click RUN to make it go. It runs the whole program every time, and you can fix the code and try again as many times as you need.

What's the difference between an object and a method?

Sample Response:

> The object is the hero, and she has methods that are things she can do. The object has a dot after it, and the method has ().

How can you tell when you've made a mistake in your code? How do you fix it?

Sample Response:

> Sometimes the code won't run because there is a mistake in it. They put a red `!` next to the mistake and try to help you. You have to read the code to figure out what's wrong.

How do comments work, and what are they for?

Sample Response:

> Comments are lines you write in the program that the computer doesn't read. If you start a line with the # symbol, the computer doesn't see it. You can write comments to remind yourself how you did something or to leave a note for another human who might want to understand your code.

Differentiation

Additional Supports:

- Show students how to find the hints, methods reference cards, error messages, and sample code provided within each level.
- Students struggling with a given level will be automatically directed to additional practice levels within the game.
- If you would like students to take notes as they work, a printable template is available here: Progress Journal
- If students struggle with breaking down problems, you can use the printable Engineering Cycle Worksheet to reinforce a step-by-step problem-solving approach.
- If students struggle to follow correct syntax, provide a copy of the printable Python Syntax Guide or JavaScript Syntax Guide

Extension Activities:

- Have students come up with a backstory for their hero. For example, why are they in the Kithgard Dungeon? What is their quest? What obstacles have they faced along their journey, before reaching the dungeon? Have them produce a written narrative, video, short play, or other creative artifact to share

their backstory with others.