Unit 6 - Python Data Types, Conditionals, Loops, and Lists

Content Area:	**CTE/Business**
Course(s):	**Intro. To Computer Science Through Gaming & Design**
Time Period:	**MarApr**
Length:	**30 Days**
Status:	**Published**

Intro to Computer Science

Department of Curriculum and Instruction

Belleville Public Schools

Curriculum Guide

Introduction to Computer Science through Gaming and Design

Unit 6: Python Data Types, Conditionals, Loops, and Lists, Grades 9-12

Belleville Board of Education

56 Ralph Street

Belleville, NJ 07109

Prepared by: Teacher, Christine Puccio

Board Approved:

Unit Overview

- This unit introduces the student to Python data types, conditionals, loops, and lists.
- There are numeric (int, float), string, and boolean data types.
- The student will notice that 2 different data types cannot be compared or combined, so casting is necessary.
- The student will see conditionals (if, if-else, if-elif, else) that create different paths of action.
- Loops will be used to repeat a statement or group of statements.
- Sometimes, nested loops or nested conditionals will be necessary to use.
- The student will make a list of elements and learn about the different list methods and functions.
- The different methods and functions will be used to alter the list or refer to specific information about the list.

Enduring Understanding

- The different data types are numeric (int, float), string, and boolean.
- Data of two different data types cannot be combined or compared without casting data.
- int, str, and float are placed around the data that needs to be casted.
- An if statement has 2 possiblities: the block inside the if does or does not happen.
- An if-else statement also has 2 possibilities: the block inside the if and the block inside the else.
- An if-elif-else statement has at least 3 possibilities.

- Each possibility between the if and else is preceded by elif, and there is no limit to the number of elifs that can be used.
- For loops repeat a statement or group of statement a specific number of times.
- To make even more possible paths of action, the loops and conditionals can be nested (placed inside each other).
- There are for i in ____ loops that iterate through the elements in a list, performing the actions for each of its items.
- Lists are placed in [ ] brackets, and the elements are separated by commas.
- There are methods and functions (append, remove, len, count, index, insert) that can alter a list or refer to its data/characteristics.
- The index indicates the position of an element, and the first one is 0.
- A while loop is used if a course of action repeats itself until a condition is met?

Essential Questions

- Why do I get an error message when I try to place "$" next to 50?
- Is casting necessary all the time?
- When does elif need to be used in a conditional?
- Where do I place each group of statements following the if, elif, and else?
- How do I know when to indent commands when I make loops?
- When I try to print the first element, why does the second one print?
- Do I need to use nested loops or nested conditionals for this program?
- How do I know which commands belong to an outer loop, and which belong to the inner loop?
- When is it convenient to use an iteration loop instead of a regular for loop?
- Are there times when only a while loop is used?

Exit Skills

At the end of Unit 6, the student should be able to:

- Determine the data type of a value.
- Understand when casting is necessary.
- Write conditional statements and determine which statements need to be indented.
- Create a list of values.
- Use the list methods and functions to alter them or refer to specific elements/information.
- Determine when nested loops and conditionals are necessary and code them correctly.

- Find the output of simple and nested loops and conditional statements.
- Find the correct index of a specific element in a given list.
- Explain when it is convenient to use a while loop instead of a for loop.

New Jersey Student Learning Standards (NJSLS-S)

CS.9-12.8.1.12.AP.1	Design algorithms to solve computational problems using a combination of original and existing algorithms.
CS.9-12.8.1.12.AP.4	Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue.
CS.9-12.8.1.12.AP.5	Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.
CS.9-12.8.1.12.CS.4	Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.
CS.9-12.8.1.12.DA.1	Create interactive data visualizations using software tools to help others better understand real world phenomena, including climate change.
CS.9-12.8.2.12.EC.3	Synthesize data, analyze trends, and draw conclusions regarding the effect of a technology on the individual, culture, society, and environment and share this information with the appropriate audience.
CS.9-12.8.2.12.ED.1	Use research to design and create a product or system that addresses a problem and make modifications based on input from potential consumers.
CS.9-12.8.2.12.ED.4	Design a product or system that addresses a global problem and document decisions made based on research, constraints, trade-offs, and aesthetic and ethical considerations and share this information with an appropriate audience.

Interdisciplinary Connections

MA.K-12.1	Make sense of problems and persevere in solving them.
MA.K-12.2	Reason abstractly and quantitatively.
MA.K-12.4	Model with mathematics.
MA.N-Q.A	Reason quantitatively and use units to solve problems.
MA.K-12.5	Use appropriate tools strategically.
MA.K-12.6	Attend to precision.
MA.K-12.7	Look for and make use of structure.
MA.K-12.8	Look for and express regularity in repeated reasoning.
LA.W.11-12.6	Use technology, including the Internet, to produce, share, and update individual or shared writing products in response to ongoing feedback, including new arguments or information.
LA.SL.11-12.4	Present information, findings and supporting evidence clearly, concisely, and logically. The content, organization, development, and style are appropriate to task, purpose, and audience.
LA.L.11-12.6	Acquire and use accurately general academic and domain-specific words and phrases,

	sufficient for reading, writing, speaking, and listening at the college and career readiness level; demonstrate independence in gathering vocabulary knowledge when considering a word or phrase important to comprehension or expression.
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.
SOC.6.1.12.A.16.a	Examine the impact of media and technology on political and social issues in a global society.
SOC.6.1.12.CS16	Contemporary United States: Interconnected Global Society: Scientific and technological changes have dramatically affected the economy, the nature of work, education, and social interactions.

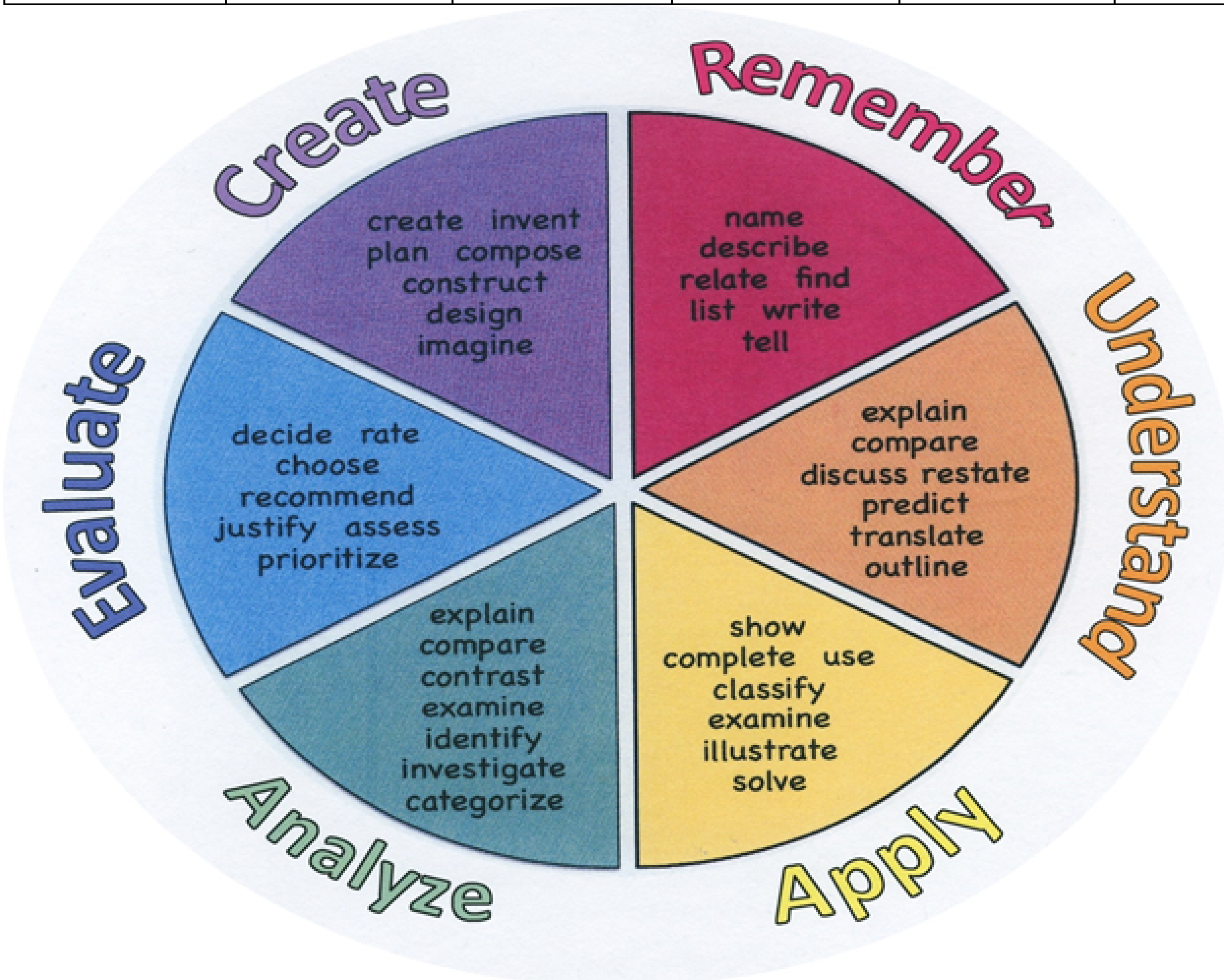
Learning Objectives

- Identify the data type of a value.
- Combine different data types by using casting.
- Design simple and nested conditionals and loops, when necessary.
- Diagram the sequence of events for simple and nested conditionals and loops.
- Revise a list by using the append, insert, and remove methods.
- Conclude and display information about a list by using the count, index, len methods/functions.
- Identify the correct element of a list by its index.
- Make code that uses a loop to iterate through a list.
- Devise a while loop and determine when it should be used in a game/animation/application.
- Determine the order of commands needed to be used to design Elsa's snowflake (Lyndsey Scott - African American)
- Design a quiz with lists that include definitions related to climate change.

Action Verbs: Below are examples of action verbs associated with each level of the Revised Bloom's Taxonomy.

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 Discuss Estimate Extrapolate Generalize Predict	Manipulate Modify Operate Subtract			



Suggested Activities & Best Practices

Best Practices:

- Short slideshow presentations for content, with questioning built into them.
- Use of multiple-choice questions to check for understanding.
- Repetition and review of concepts.
- Immediate feedback for assignments.
- Step by step visual instructions to make programs, especially at the beginning.
- The use of EduBlocks, where blocks appear in one panel, but the Python code appears in the other panel.
- Show model code for students to refer to.
- Provide example output for students to compare their results.

- Google Classroom and Schoology organized around units of study.

Exemplars:

- Have examples of Python code blocks and code available for students to use as a model.
- Use questioning activities about Python turtle where students get feedback after submission (Edulastic, Google Forms, quizizz.com).
- Use slideshow notes for instructions to use Python, with illustrations of the blocks in English and Spanish.
- Have students continue to apply previous Python turtle concepts when using other Python concepts.

Assessment Evidence - Checking for Understanding (CFU)

- edulastic.com - for practice exercises and assessment (Formative and Summative)
- whiteboard.fi/ - to present notes and questions (Formative)
- Jamboard - for group work (Formative)
- Google Forms - for Do Nows, Exit Tickets and Assessment activities (Formative)
- Coding Rooms - for practice programs and projects (Formative)
- EduBlocks - for block-based python programming on one side and python code on the other (Formative/Summative)

Performance Task Example (Alternate):

Go through websites on African American contributions to coding.
Make a brief program with Python conditionals, codes, and lists with the facts that were found.
You may choose any article or video that was not listed in the suggested articles/websites.

- Google Slides - for Notes and Drag and Drop activities (Formative)
- Google Classroom - for open-ended questions (Formative)
- quizizz.com - for content practice in a game format (Alternate)
- oncourse.com - for benchmarks (if applicable) (Summative/Benchmark)

- Admit Tickets
- Anticipation Guide
- Common Benchmarks
- Compare & Contrast
- Create a Multimedia Poster
- Define
- Describe
- Evaluate
- Evaluation rubrics
- Exit Tickets
- Explaining
- Illustration
- Learning Center Activities
- Multimedia Reports
- Outline
- Quizzes

- Self- assessments
- Study Guide
- Teacher Observation Checklist
- Think, Pair, Share
- Think, Write, Pair, Share
- Unit review/Test prep
- Unit tests
- Web-Based Assessments
- Written Reports

Primary Resources & Materials

- Creative Coding in Python
- Coding for Kids Python
- Coding Games in Python
- https://edublocks.org/

Ancillary Resources

Coding:

- https://www.codingrooms.com/
- https://trinket.io/python
- https://realpython.com/beginners-guide-python-turtle/
- https://blockly.games/
- https://www.youtube.com/c/TechWithTim/search?query=turtle (Tech with Tim Python Turtle Graphics Tutorials)
- https://hourofpython.com/a-visual-introduction-to-python/index.html
- https://studio.code.org/s/csp4-2021/lessons/5/levels/1 (variables, conditionals, and loops)
- https://compute-it.toxicode.fr/?hour-of-code&progression=python

African Americans:

- https://studio.code.org/s/frozen/lessons/1/ (Coding with Elsa - Lyndsey Scott)
- https://www.codecademy.com/resources/blog/black-programmers-and-technologists-who-inspire-us/
- https://www.infoworld.com/article/3606070/hidden-figures-7-black-programmers-you-should-know.html
- https://www.hackreactor.com/blog/black-software-engineers-throughout-history

Climate Change:

- https://climate.nasa.gov/

- https://www.epa.gov/climate-change
- https://education.nationalgeographic.org/resource/climate-change

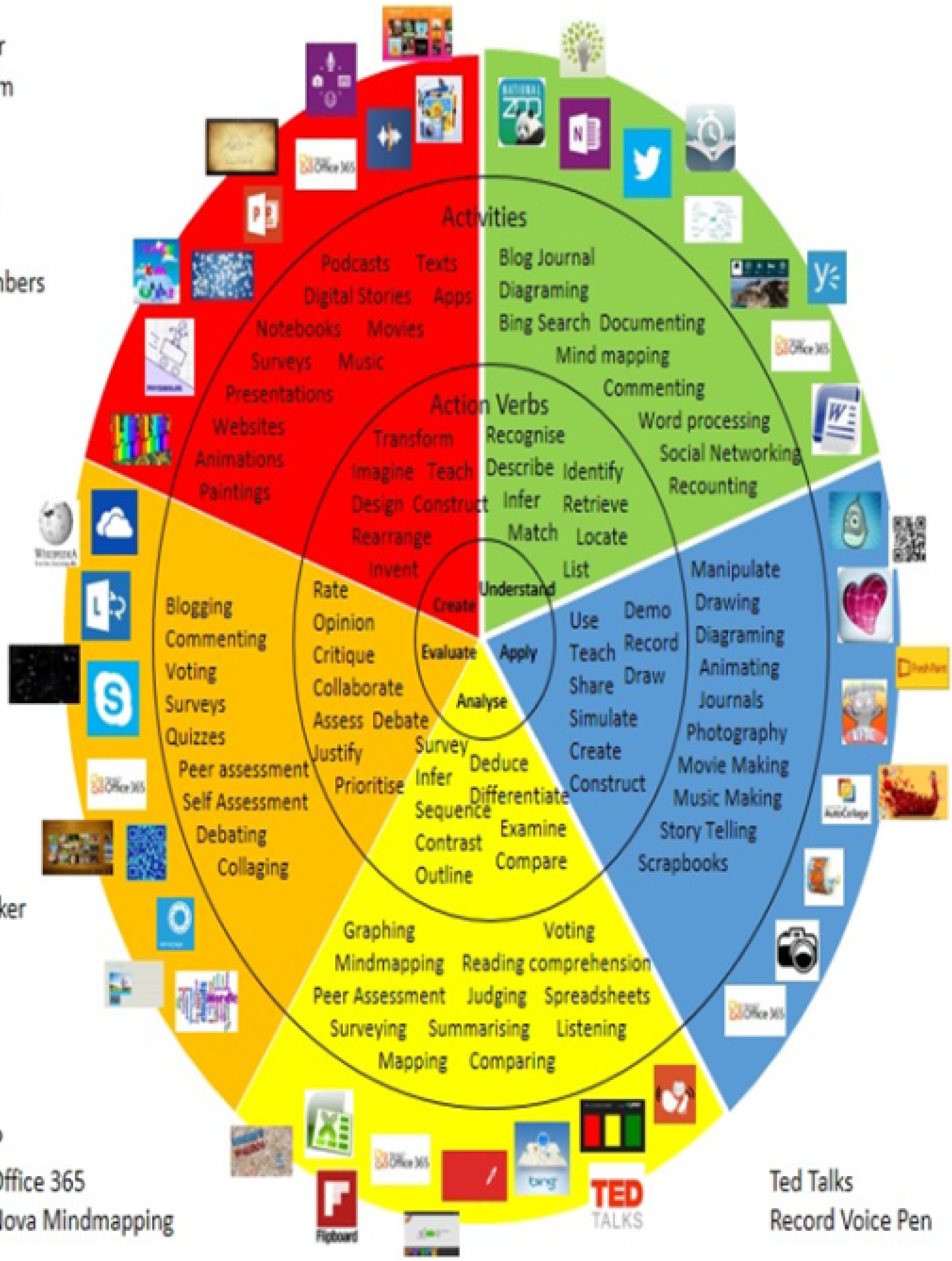
Technology Infusion

- use of the internet - for Python websites and articles about climate change and African Americans
- edulastic.com - for practice exercises and assessment
- whiteboard.fi/ - to present notes and questions
- Jamboard - for group work
- Google Forms - for Do Nows, Exit Tickets and Assessment activities
- Google Slides - for Notes and Drag and Drop activities (Formative)
- Google Classroom - for open-ended questions (Formative)
- quizizz.com - for content practice in a game format (Alternate)
- edublocks.org - for programs and games (Formative/Summative)
- trinket.io/python - for programs (Formative)
- https://www.codingrooms.com/ (Formative)
- oncourse.com - for benchmarks (if applicable) (Summative/Benchmark)

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
Flipboard
Office 365
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

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

WRK.9.2.12.CAP.2	Develop college and career readiness skills by participating in opportunities such as structured learning experiences, apprenticeships, and dual enrollment programs.
WRK.9.2.12.CAP.6	Identify transferable skills in career choices and design alternative career plans based on those skills.
TECH.9.4.12.CI.1	Demonstrate the ability to reflect, analyze, and use creative skills and ideas (e.g., 1.1.12prof.CR3a).
TECH.9.4.12.CI.3	Investigate new challenges and opportunities for personal growth, advancement, and transition (e.g., 2.1.12.PGD.1).

21st Century Skills/Interdisciplinary Themes

Exemplars:

- Students utilize different types of software for activities.
- Students identify the different parts (conditionals, variables, loops, lists, etc.) of a program.

- 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

Exemplars:

- Students acknowledge technological contributions of African Americans.
- Students are inspired to develop applications for personal or business purposes.

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

Differentiation

Differentiations:

- Small group instruction
- Small group assignments
- Extra time to complete assignments
- Pairing oral instruction with visuals
- Repeat directions
- Use manipulatives
- Center-based instruction
- Study guides
- Teacher reads assessments allowed
- Scheduled breaks
- Rephrase written directions
- Multisensory approaches
- Additional time
- Preview vocabulary
- Preview content & concepts
- Behavior management plan
- Highlight text
- Student(s) work with assigned partner
- Visual presentation
- Assistive technology
- Auditory presentations
- Small group setting

Hi-Prep Differentiations:

- Alternative formative and summative assessments
- Games and tournaments
- Group investigations
- Guided Reading
- Independent research and projects
- Learning contracts
- Leveled rubrics
- Multiple intelligence options
- Multiple texts
- Project-based learning
- Problem-based learning
- Stations/centers
- Tiered activities/assignments

- Tiered products
- Varying organizers for instructions

Lo-Prep Differentiations

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

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

Exemplars:

- Allow multiple-choice assignments, written assignments, and quizzes to be submitted late.
- Convert articles to PDF and highlight important ideas for students.
- Make visual instructions in addition to the ones included in the software.
- Give students the opportunity to unscramble computer commands instead of generating their own.

- 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:

- Have all notes, activity directions, and assessment items translated into Spanish.
- Place students next to Spanish-speaking peers.
- Make visual instructions.
- Have individual interaction with students to make sure that they understand the content and expectations.
- Allow students to use the drop-down menu to choose their native language on software, when applicable.

- teaching key aspects of a topic. Eliminate nonessential information.
- using videos, illustrations, pictures, and drawings to explain or clarify
- 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 work presented 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

Exemplars:

- Minimize the amount of reading that needs to be done.
- Minimize the amount of information that students need to write/type.
- Make visual instructions in addition to the ones provided in the software.
- When asking questions, give students possible answers to choose from.
- Give students the opportunity to unscramble commands instead of having to generate them by themselves.

- 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 work presented 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:

- Have students do further research on climate change and African American contributions to technology.
- Allow students to make their own applications beyond the classroom assignments.
- Let students see their Python program with blocks converted to actual Python code.
- Encourage students to try typing their own Python code.

- 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 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

Unit Name: Lyndsey Scott (an African American programmer) and the Frozen Activity

NJSLS:

Interdisciplinary Connection: Social Studies connection - Students see the contributions of American Americans to technology.

Statement of Objective: The student should be able to:

- Name African Americans that contributed to technology.
- Use Python code to design a snowflake.

Anticipatory Set/Do Now: Watch Lyndsey Scott (as Elsa from Frozen) on code.org

Learning Activity: Do Now.

Read articles about Lyndsey Scott and other African American programmers.

Whole-class discussion about African American programmers.

Continue Elsa's Frozen activity to design the snowflake.

Student Assessment/CFU's: observation, questioning

Materials: computer/Chromebook, articles about African American programmers, internet access to code.org

21st Century Themes and Skills: communication, critical thinking, information literacy

Differentiation/Modifications: provide extra directions and code to supplement the software, peer tutoring

Integration of Technology: use of computer/Chromebook, use of internet for code.org and articles

CS.9-12.8.1.12.AP.1	Design algorithms to solve computational problems using a combination of original and existing algorithms.
CS.9-12.8.2.12.EC.3	Synthesize data, analyze trends, and draw conclusions regarding the effect of a technology on the individual, culture, society, and environment and share this information with the

	appropriate audience.
CS.9-12.8.2.12.ED.1	Use research to design and create a product or system that addresses a problem and make modifications based on input from potential consumers.
SOC.6.2.12.D.6.a	Assess the role of increased personal and business electronic communications in creating a “global” culture, and evaluate the impact on traditional cultures and values.
WRK.9.2.12.CAP.2	Develop college and career readiness skills by participating in opportunities such as structured learning experiences, apprenticeships, and dual enrollment programs.
WRK.9.2.12.CAP.6	Identify transferable skills in career choices and design alternative career plans based on those skills.
TECH.9.4.12.CI.1	Demonstrate the ability to reflect, analyze, and use creative skills and ideas (e.g., 1.1.12prof.CR3a).
TECH.9.4.12.CI.3	Investigate new challenges and opportunities for personal growth, advancement, and transition (e.g., 2.1.12.PGD.1).