Unit 6 Inheritance

Content Area: **CTE**
Course(s): **Sample Course**
Time Period:
Length: **20 class periods, Grades 10-12**
Status: **Published**

AP Computer Science A, Inheritance

Department of Curriculum and Instruction

Belleville Public Schools

Curriculum Guide

AP COMPUTER SCIENCE A

Inheritance

Belleville Board of Education

102 Passaic Avenue

Belleville, NJ 07109

Prepared by: Teacher, Corey Woodring

Unit Overview

Inheritance in Java is not an estate or a classic car from a long-lost relative. It means certain classes can share attributes from other classes. In this unit we'll learn the concept and the syntax for this powerful feature of Java.

Inheritance is a wonderful thing in Java. It's a term used often in object-oriented programming. But what does it really mean? And how do you harness this powerful concept?

To **inherit** in Java means allowing all methods and variables from one class to be accessible by another class. That is, the new class inherits these items. The **parent class**, also called superclass, is the class whose methods and variables can be used in the **child class** (also called subclass).

Another way to think of the inheritance concept is the phrase: A union employee is an employee; a paperback book is a book. It's more than a child inheriting 50% of their parent's DNA - they get all of it. The subclass gets all the goodies from the parent class, but it can also create its own variables and methods.

Inheritance is a key part of object-oriented programming. It allows for use and re-use of objects, methods, and variables, without having to add extra/redundant code.

Creating objects, calling methods on the objects created, and being able to define a new data type by creating a class are essential understandings before moving into this unit. One of the strongest advantages of Java is the ability to categorize classes into hierarchies through inheritance. Certain existing classes can be extended to

include new behaviors and attributes without altering existing code. These newly created classes are called subclasses. In this unit, students will learn how to recognize common attributes and behaviors that can be used in a superclass and will then create a hierarchy by writing subclasses to extend a superclass. Recognizing and utilizing existing hierarchies will help students create more readable and maintainable programs.

Students can design hierarchies by listing the attributes and behaviors for each object and pulling common elements into a superclass, leaving unique attributes and behaviors in the subclass. By creating a hierarchy system, students only need to write common program code one time, reducing potential errors and implementation time. This also allows for changes to be made more easily, because they can be made at the superclass level in the hierarchy and automatically apply to subclasses. During the development of a program, programmers often use comments to describe the behavior of a given segment of program code and to describe the initial conditions that are used. Students who develop the skill of explaining why program code does not work, such as methods being overloaded improperly or superclass objects attempting to call subclass methods, are much better equipped to foresee and avoid these hierarchy issues.

Enduring Understanding

When multiple classes contain common attributes and behaviors, programmers create a new class containing the shared attributes and behaviors forming a hierarchy.

Modifications made at the highest level of the hierarchy apply to the subclasses.

Essential Questions

How might the use of inheritance help in writing a program that simulates crops being grown in a virtual world?

How does inheritance make programs more versatile?

Exit Skills

Determine an appropriate program design to solve a problem or accomplish a task (not assessed). 3.B Write program code to define a new type by creating a class. 9.2 Writing Constructors for Subclasses 3.B Write program code to define a new type by creating a class. 5.A Describe the behavior of a given segment of program code. 9.3 Overriding Methods 3.B Write program code to define a new type by creating a class. 5.D Describe the initial conditions that must be met for a program segment to work as intended or described. 9.4 super Keyword 1.C Determine code that would be used to interact with completed program code. 3.B Write program code to define a new type by creating a class. 9.5 Creating References Using Inheritance Hierarchies 3.A Write program code to create objects of a class and call methods. 5.B Explain why a code segment will not compile or work as intended. 9.6 Polymorphism 3.A Write program code to create objects of a class and call methods. 5.B Explain why a code segment will not compile or work as intended. 9.7 Object Superclass 1.C Determine code that would be used to interact with completed program code. 3.B Write program code to define a new type by creating a class

New Jersey Student Learning Standards (NJSLS-S)

CSTA Standards
Computing Systems • 3A-CS-03 Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors. • 3B-CS-02 Illustrate ways computing systems implement logic, input, and output through hardware components. ***Networks and the Internet*** • 3B-NI-03 Describe the issues that impact network functionality (e.g., bandwidth, load, delay, topology). ***Data and Analysis*** • 3A-DA-09 Translate between different bit representations of real-world phenomena, such as characters, numbers, and images. • 3A-DA-12 Create computational models that represent the relationships among different elements of data collected from a phenomenon or process.

- 3B-DA-06 Select data collection tools and techniques to generate data sets that support a claim or communicate information.
- 3B-DA-07 Evaluate the ability of models and simulations to test and support the refinement of hypotheses.

Algorithms and Programming

- 3A-AP-13 Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.
- 3A-AP-15 Justify the selection of specific control structures when tradeoffs involve implementation, readability, and program performance, and explain the benefits and drawbacks of choices made.
- 3A-AP-17 Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.
- 3A-AP-19 Systematically design and develop programs for broad audiences by incorporating feedback from users.
- 3A-AP-21 Evaluate and refine computational artifacts to make them more usable and accessible.
- 3A-AP-22 Design and develop computational artifacts working in team roles using collaborative tools.
- 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-14 Construct solutions to problems using student-created components, such as procedures, modules and/or objects.
- 3B-AP-20 Use version control systems, integrated development environments (IDEs), and collaborative tools and practices (code documentation) in a group software project.
- 3B-AP-21 Develop and use a series of test cases to verify that a program performs according to its design specifications.
- 3B-AP-22 Modify an existing program to add additional functionality and discuss intended and unintended implications (e.g., breaking other functionality).
- 3B-AP-23 Evaluate key qualities of a program through a process such as a code review.

Impacts of Computing

- 3A-IC-25 Test and refine computational artifacts to reduce bias and equity deficits.
- 3A-IC-26 Demonstrate ways a given algorithm applies to problems across disciplines.
- 3A-IC-29 Explain the privacy concerns related to the collection and generation of data through automated processes that may not be evident to users.

8.2.12.B.1 Research and analyze the impact of the design constraints (specifications and limits) for a product or technology driven by a cultural, social, economic or political need and publish for review. 8.2.12.B.3 Analyze ethical and unethical practices around intellectual property rights as influenced by human wants and/or needs. 8.2.12.C.1 Explain how open source technologies follow the design process. 8.2.12.E.1 Demonstrate an understanding of the problem-solving capacity of computers in our world. 8.2.12.E.2 Analyze the relationships between internal and external computer components. 8.2.12.E.3 Use a programming language

to solve problems or accomplish a task (e.g., robotic functions, website designs, applications, and games). 8.2.12.E.4 Use appropriate terms in conversation (e.g., troubleshooting, peripherals, diagnostic software, GUI, abstraction, variables, data types and 8.2.12.B.1 Research and analyze the impact of the design constraints (specifications and limits) for a product or technology driven by a cultural, social, economic or political need and publish for review. 8.2.12.B.3 Analyze ethical and unethical practices around intellectual property rights as influenced by human wants and/or needs. 8.2.12.C.1 Explain how open source technologies follow the design process. 8.2.12.E.1 Demonstrate an understanding of the problemsolving capacity of computers in our world. 8.2.12.E.2 Analyze the relationships between internal and external computer components. 8.2.12.E.3 Use a programming language to solve problems or accomplish a task (e.g., robotic functions, website designs, applications, and games). 8.2.12.E.4 Use appropriate terms in conditional statements). Support Standards (CSTA): CPP.L2.04 Demonstrate an understanding of algorithms and their practical application CPP.L3A.05 Use Application Programming Interfaces (APIs) and libraries to facilitate programming solutions. CPP.L3A.08 Explain the program execution process. CRP Standards: 9.2.12.C.3 Identify transferable career skills and design alternate career plans. 9.2.12.C.5 Research career opportunities in the United States and abroad that require knowledge of world languages and diverse cultures. 9.2.12.C.6 Investigate entrepreneurship opportunities as options for career planning and identify the knowledge, skills, abilities, and resources required for owning and managing a business. 9.2.12.C.7 Examine the professional, legal, and ethical responsibilities for both employers and employees in the global workplace. CRP2 Apply appropriate academic and technical skills. CRP10 Plan education and career paths aligned to personal goals. CRP11 Use technology to enhance conditional statements).

CPP.L2.04 Demonstrate an understanding of algorithms and their practical application CPP.L3A.05 Use Application Programming Interfaces (APIs) and libraries to facilitate programming solutions. CPP.L3A.08 Explain the program execution process. CRP Standards: 9.2.12.C.3 Identify transferable career skills and design alternate career plans. 9.2.12.C.5 Research career opportunities in the United States and abroad that require knowledge of world languages and diverse cultures. 9.2.12.C.6 Investigate entrepreneurship opportunities as options for career planning and identify the knowledge, skills, abilities, and resources required for owning and managing a business. 9.2.12.C.7 Examine the professional, legal, and ethical responsibilities for both employers and employees in the global workplace. CRP2 Apply appropriate academic and technical skills. CRP10 Plan education and career paths aligned to personal goals. CRP11 Use technology to enhance.

TECH.8.2.12.D.4	Assess the impacts of emerging technologies on developing countries.
TECH.8.2.12.D.6	Synthesize data, analyze trends and draw conclusions regarding the effect of a technology on the individual, society, or the environment and publish conclusions.
TECH.8.2.12.E.1	Demonstrate an understanding of the problem-solving capacity of computers in our world.
TECH.8.2.12.E.2	Analyze the relationships between internal and external computer components.
TECH.8.2.12.E.3	Use a programming language to solve problems or accomplish a task (e.g., robotic functions, website designs, applications, and games).

Interdisciplinary Connections

Primary interdisciplinary connections:

Infused within the unit are connections to the 2009 NJCCCS for Language Arts Literacy and Business, Science and Technology.

Critical reading, writing, and mathematical modeling skills are promoted within the problem solving process and as a means to explain solutions.

	Integration of Knowledge and Ideas
LA.RST.11-12.3	Follow precisely a complex multistep procedure when carrying out experiments, taking measurements, or performing technical tasks; analyze the specific results based on explanations in the text.
	The set of inputs to a function is called its domain. We often infer the domain to be all inputs for which the expression defining a function has a value, or for which the function makes sense in a given context.

Learning Objectives

Students will learn how to recognize common attributes and behaviors that can be used in a superclass and will then create a hierarchy by writing subclasses to extend a superclass. Recognizing and utilizing existing hierarchies will help students create more readable and maintainable programs.

Suggested Activities & Best Practices

1 Activating prior knowledge Have students review what they know about classes, methods, and the scope of variables by having them write a class based on specifications that can easily be extended by subclasses. This class will become the superclass for subclasses they write later in the unit. 2 9.2–9.4 Create a plan Given a class design problem that requires the use of multiple classes in an inheritance hierarchy, students identify the common attributes and behaviors among these classes and write these into a superclass. Any additional information that does not belong in the superclass will be categorized to determine the additional classes that might be necessary and what methods will need to be added or overridden in the subclasses. 3 9.4 Think aloud Provide students with a code segment that contains method calls using the super keyword. Have students describe the code segment out loud to themselves. Give students several individual statements that attempt to interact with the given code segment, and have them talk through each one, describing which statements would work and which ones would not, as well as the reasons why those statements wouldn't work. 4 9.5–9.6 Student response system Provide students with several statements where objects are created and the reference type and object type are different but related. Then provide students with calls to methods on these created objects. Use a student response system to have students determine whether each statement is legal, would result in a compile-time error, or would result in a run-time error.

Assessment Evidence - Checking for Understanding (CFU)

- Complete various written checkpoint exercises that focus on the explanation and description of computer hardware and Java basics.-alternate assessment
- Develop a visual representation of the communication processes within a computer using appropriate terminology.
- Properly document a program using correct indentation, spacing, and comment style.
- Debug programs and determine the types of errors in the program.
- Create programs based on programming exercises that display various types of output using string and numeric expressions.--Benchmark Assessment

Formative Evaluations:	**Summative Evaluations:**
Formative Assessment with polling codeIt! Nows Quizzes Long Programs (LP)/Lab Work Components of AP approved Lab	Unit Test/ReTest-Summative Assessment

Evidence of Student Learning with Checking for Understanding (CFU) techniques used during the lesson and/or for Closure (Madeline Hunter), please list the variety of means used to access students' learning (e.g. quizzes, tests, academic prompts, observations, homework, journals).

- 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

NJCTL Curriculum

Curriculum development Resources/Instructional Materials/Equipment Needed Teacher Resources: • www.gliffy.com • Eclipse IDE • MS DOS Prompt • Computers

Ancillary Resources

Java Resources

- Java Review for the AP CS A Exam - Great review site with lots of practice questions.
- Aplus Compter Science Exam Review Material -Slide, Free Response, and more!
- Introduction to Java - a textbook for a first course in computer science for the next generation of scientists and engineers
- Guru-99 Introduction Java Material
- Oracles (owners of Java) has their own tutorials
- Dick Baldwin - ACC - Introduction and Advanced Java Material
- Introduction to Computer Science using Java - by Bradley Kjell
- Thinking in Java
- Blue Pelican Java
- Java Coding Bat - Lots of good practice problems
- Code Academy - No Java but good practice.

https://chortle.ccsu.edu/CS151/cs151java.html

Technology Infusion

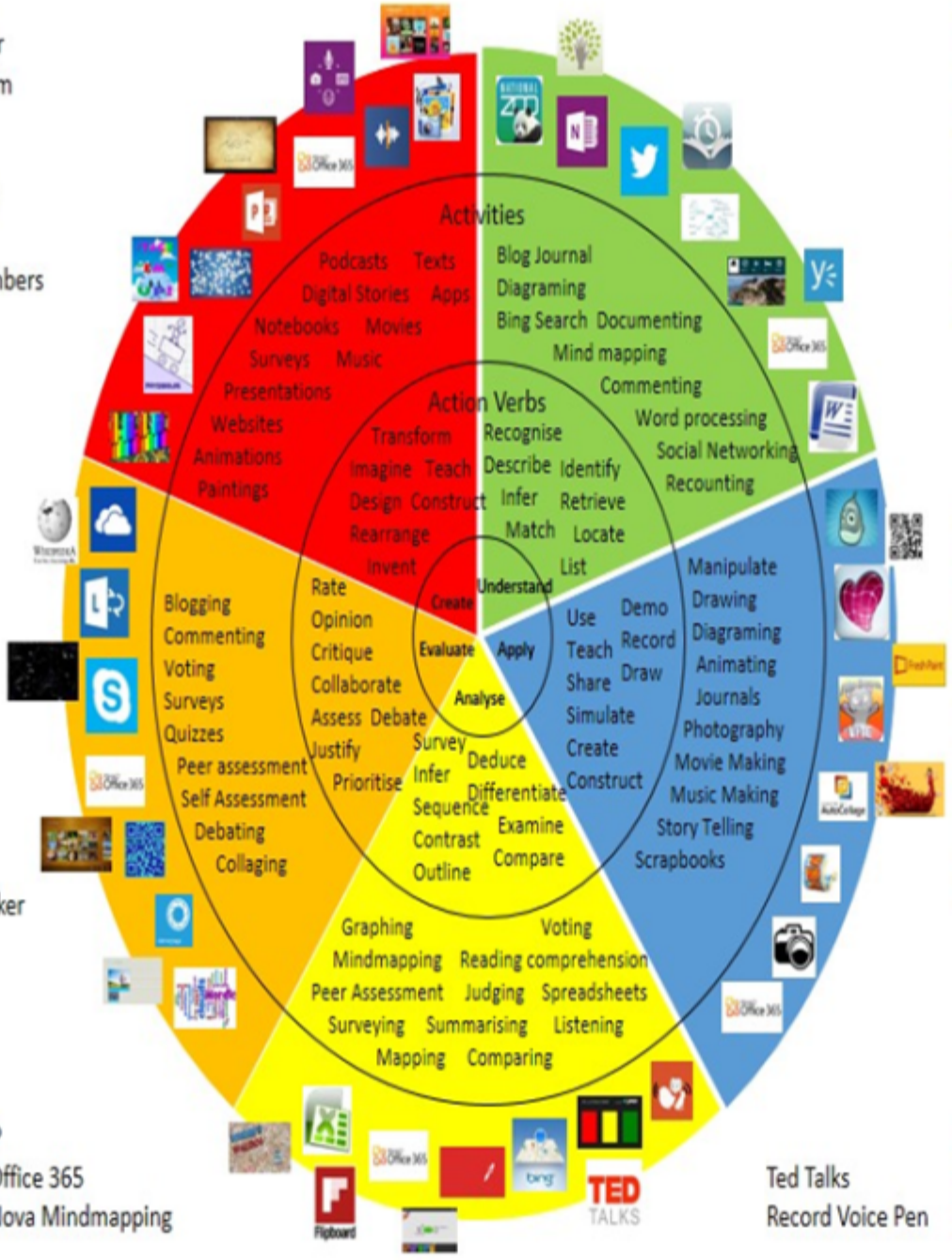
Technology Infusion and/or strategies include chromebooks online materials google/powerpoint slides

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

21st century themes: The unit will integrate the 21st Century Life and career standard 9.1 strands A-D. These strands include: critical thinking and problem solving, creativity and innovation, collaboration, teamwork, and leadership, and cross cultural understanding and interpersonal communication

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.

Functions

Functions presented as expressions can model many important phenomena. Two important families of functions characterized by laws of growth are linear functions, which grow at a constant rate, and exponential functions, which grow at a constant percent rate. Linear functions with a constant term of zero describe proportional relationships.

The set of inputs to a function is called its domain. We often infer the domain to be all inputs for which the expression defining a function has a value, or for which the function makes sense in a given context.

21st Century Skills/Interdisciplinary Themes

21st century themes: The unit will integrate the 21st Century Life and career standard 9.1 strands A-D. These strands include: critical thinking and problem solving, creativity and innovation, collaboration, teamwork, and leadership, and cross cultural understanding and interpersonal communication

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

TECH.K-12.1.2	Digital Citizen
TECH.K-12.1.5	Computational Thinker
TECH.K-12.1.5.a	formulate problem definitions suited for technology-assisted methods such as data analysis, abstract models and algorithmic thinking in exploring and finding solutions.
TECH.K-12.1.5.b	collect data or identify relevant data sets, use digital tools to analyze them, and represent data in various ways to facilitate problem-solving and decision-making.
TECH.K-12.1.5.c	break problems into component parts, extract key information, and develop descriptive models to understand complex systems or facilitate problem-solving.
TECH.K-12.1.5.d	understand how automation works and use algorithmic thinking to develop a sequence of steps to create and test automated solutions.

21st Century Skills

21st Century Skills that will be incorporated into this unit.

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

CAEP.9.2.12.C	Career Preparation
CAEP.9.2.12.C.1	Review career goals and determine steps necessary for attainment.
CAEP.9.2.12.C.2	Modify Personalized Student Learning Plans to support declared career goals.
CAEP.9.2.12.C.3	Identify transferable career skills and design alternate career plans.

Differentiation

• Technology Resources • Teacher Tutoring • Peer Tutoring • Cooperative Learning Groups • Differentiated Instruction • Follow all IEP Modifications/504 Plan

Exemplar: Pairing oral instruction with visuals

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)

Exemplar:

Students work with an assigned partner.

Weekly Assignment Sheet in Google Classroom

Adapting existing materials, simplifying or supplementing materials

- **Adjust** the method of presentation or content.
- **Develop** supplemental material.
- **Tape-record** directions for the material.
- **Provide** alternatives for responding to questions.
- **Rewrite** brief sections to lower the reading level.
- **Outline** the material for the student before reading a selection.
- **Reduce** the number of pages or items on a page to be completed by thestudent.
- **Break** tasks into smaller subtasks.
- **Provide** additional practice to ensure mastery.
- **Substitute** a similar, less complex task for a particular assignment.
- **Develop** simple study guides to complement required materials.

Special Education Learning adaptations that will be employed in the unit, using the ones identified below.

- 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)

Exemplar: Give simplified written and verbal instructions

Whether you're giving instructions for a test or assignment verbally or textually, you should simplify them as much as possible for students with LEP. Many times, this can be easily done by taking out extra words or turning complex sentences into simple ones.

- 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

Exemplar:

Directions or Instructions

Make sure directions and/or instructions are given in limited numbers. Give directions/instructions verbally and in simple written format. Ask students to repeat the instructions or directions to ensure understanding occurs. Check back with the student to ensure he/she hasn't forgotten.

- 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)

Exemplar:

Students will create a blog or social media page on a topic of their choice within the unit

http://www.grandviewlibrary.org/CurriculumAdaptations/General_Gifted.pdf

Grouping • Group gifted students with other gifted students or higher-level learners. • Refrain from grouping gifted students with lower-level students for remediation.

- 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://runestone.academy/runestone/books/published/csawesome/Unit9-Inheritance/toctree.html?highlight=inheritance

Using the template below, please develop a **Sample Lesson** for the first unit only.

Unit Name:

NJSLS:

Interdisciplinary Connection:

Statement of Objective:

Anticipatory Set/Do Now:

Learning Activity:

Student Assessment/CFU's:

Materials:

21st Century Themes and Skills:

Differentiation/Modifications:

Integration of Technology: