Unit 2: Programming Copied from: Computer Science - Principles, Copied on: 02/21/22

Content Area: **Technology**
Course(s): **Sample Course, AP Computer Science Principles**
Time Period: **OctNov**
Length: **Sample Length & Grade Level**
Status: **Published**

Computer Science Principles

Department of Curriculum and Instruction

Belleville Public Schools

Curriculum Guide

Computer Science Principles, Grades 9-12

Unit 2, Programming

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 K-8, ESL Coordinator K-12

Mr. George Droste, Director of Secondary Education

Board Approved: September 23, 2019

Unit Overview

Use of a programming environment to explore sequencing, selection, and iteration as part of the goal to create programs that serve useful functions.

Enduring Understanding

- Computing facilitates exploration and the creation of computational artifacts and new knowledge that help people solve personal, societal, and global problems.
- Computing allows people to use creative development processes to produce computational artifacts for creative expression or to solve a problem.
- Computing has global effects both beneficial and harmful on people and society.

Essential Questions

- How are programs developed to help people, organizations, or society solve problems?
- How does abstraction make the development of computer programs possible?

Exit Skills

Students will be able to:

Visual Programming:

- Utilize a graphical editor to read, construct, and execute dynamic programs.
- Assess, modify, and execute programs developed by others.
- Examine how well-specified behavior of objects can be constructed through sequential actions and operations.

Program State:

- Develop a variety of programs using methods and techniques that are appropriate for the goals of the programmer.
- Create programs that incorporate dynamic, user-driven, keyboard controls and input.
- Experiment with how the dynamic state of an object or program can be stored and changed using variables.
- Analyze the role of clear, descriptive names for objects, behaviors, variables, and other identifiers in maintaining the readability of code.
- Identify additional desired outcomes for a program that extend beyond its original purpose.

Selection Statements:

- Examine the uses of selection statements in programming.
- Analyze the differences between simple selection and complex, nested selection statements.
- Constructing complex conditional statements with the use of the Boolean operators "AND", "OR", and "NOT" in coding.

Coding Skills:

- Explain how algorithms are implemented using program instructions that are processed sequentially during program execution.
- Design and construct instructions using a non-traditional, domain specific notation.
- Evaluate the clarity and legibility of instructions written in a nontraditional, domain-specific notation by reading and executing instructions created by others.
- Examine a number of common programming errors.
- Explore a number of common debugging strategies.
- Develop solution and strategies for correcting common programming errors.

Global Impact:

- Examine and discuss the motivations behind a number of high profile individuals in the field of programming.
- Discuss the benefits of programming as a tool and a profession.

Analyze the legal and ethical concerns of open source and licensed software, libraries, and code.

New Jersey Student Learning Standards (NJSLS-S)

TECH.8.1.12	Educational Technology: All students will use digital tools to access, manage, evaluate, and synthesize information in order to solve problems individually and collaborate and to create and communicate knowledge.
TECH.8.1.12.A	Technology Operations and Concepts: Students demonstrate a sound understanding of technology concepts, systems and operations.
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	Computational Thinking: Programming: Computational thinking builds and enhances problem solving, allowing students to move beyond using knowledge to creating knowledge.
TECH.8.2.12.E.1	Demonstrate an understanding of the problem-solving capacity of computers in our world.
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).
TECH.8.2.12.E.4	Use appropriate terms in conversation (e.g., troubleshooting, peripherals, diagnostic software, GUI, abstraction, variables, data types and conditional statements).
TECH.8.2.12.E.CS1	Computational thinking and computer programming as tools used in design and engineering.

Interdisciplinary Connections

21st Century Life and Careers

9.1.12.B.1, 9.1.12.F.2

9.3.12.K3.2, 9.3.12.K3.4, 9.3.12.K3.5, 9.3.12.K3.6

21st century life and careers

Technology

Digital Literacy

English Language Arts

Reading

Writing

Speaking and Listening (communication skills)

Mathematics

Social science

Science and the Engineering Practices

Career Ready Practices

CRP4. Communicate clearly and effectively and with reason.

CRP5. Consider the environmental, social and economic impacts of decisions.

CRP6. Demonstrate creativity and innovation.

CRP7. Employ valid and reliable research strategies.

CRP8. Utilize critical thinking to make sense of problems and persevere in solving them.

CRP11. Use technology to enhance productivity.

LA.RST.6-8.1	Cite specific textual evidence to support analysis of science and technical texts.
LA.RST.6-8.3	Follow precisely a multistep procedure when carrying out experiments, taking measurements, or performing technical tasks.
LA.WHST.6-8.1.A	Introduce claim(s) about a topic or issue, acknowledge and distinguish the claim(s) from alternate or opposing claims, and organize the reasons and evidence logically.
LA.WHST.6-8.1.C	Use words, phrases, and clauses to create cohesion and clarify the relationships among claim(s), counterclaims, reasons, and evidence.
LA.WHST.6-8.10	Write routinely over extended time frames (time for research, reflection, metacognition/self correction, and revision) and shorter time frames (a single sitting or a day or two) for a range of discipline-specific tasks, purposes, and audiences.
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-ETS1-1	Analyze a major global challenge to specify qualitative and quantitative criteria and

constraints for solutions that account for societal needs and wants.

Modeling links classroom mathematics and statistics to everyday life, work, and decision-making. Modeling is the process of choosing and using appropriate mathematics and statistics to analyze empirical situations, to understand them better, and to improve decisions. Quantities and their relationships in physical, economic, public policy, social, and everyday situations can be modeled using mathematical and statistical methods. When making mathematical models, technology is valuable for varying assumptions, exploring consequences, and comparing predictions with data.

Learning Objectives

Develop a variety of programs using methods and techniques that are appropriate for the goals of the programmer.

- Create programs that incorporate dynamic, user-driven, keyboard controls and input.
- Experiment with how the dynamic state of an object or program can be stored and changed using variables.
- Analyze the role of clear, descriptive names for objects, behaviors, variables, and other identifiers in maintaining the readability of code.
- Identify additional desired outcomes for a program that extend beyond its original purpose.

Suggested Activities & Best Practices

Visual Programming:

- Utilize a graphical editor to read, construct, and execute dynamic programs.
- Assess, modify, and execute programs developed by others.
- Examine how well-specified behavior of objects can be constructed through sequential actions and operations.

Program State:

Develop a variety of programs using methods and techniques that are appropriate for the goals of the programmer.

- Create programs that incorporate dynamic, user-driven, keyboard controls and input.
- Experiment with how the dynamic state of an object or program can be stored and changed using variables.

- Analyze the role of clear, descriptive names for objects, behaviors, variables, and other identifiers in maintaining the readability of code.
- Identify additional desired outcomes for a program that extend beyond its original purpose.

Selection Statements:

- Examine the uses of selection statements in programming.
- Analyze the differences between simple selection and complex, nested selection statements.
- Constructing complex conditional statements with the use of the Boolean operators "AND", "OR", and "NOT" in coding.

Coding Skills:

- Explain how algorithms are implemented using program instructions that are processed sequentially during program execution.
- Design and construct instructions using a non-traditional, domain specific notation.
- Evaluate the clarity and legibility of instructions written in a nontraditional, domain-specific notation by reading and executing instructions created by others.
- Examine a number of common programming errors.
- Explore a number of common debugging strategies.
- Develop solution and strategies for correcting common programming errors.

Global Impact:

- Examine and discuss the motivations behind a number of high profile individuals in the field of programming.
- Discuss the benefits of programming as a tool and a profession.
- Analyze the legal and ethical concerns of open source and licensed software, libraries, and code.

Assessment Evidence - Checking for Understanding (CFU)

Formative Assessments

• Think, pair, share review questions from text.

• Practice mini-programs to strengthen concepts as taught.
• Teacher Observation

• Utilizing Gliffy.com to flowchart programs
Summative Assessments

• Chapter Test

• End of Chapter Projects from book.

example: https://quizlet.com/76598385/ap-computer-science-a-flash-cards/

Alternate Assessment

Web-based Assessment

Benchmark Assessment

Develop solution and strategies for correcting common programming errors.

- 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

Primary Resources:

Edhesive Online Training
Code.Org Online Training

Computers and Internet Access
AP Central at Collegeboard.org
Massive Open Online Course
Code.org
Multimedia Applications Tools
Abelson, H., Ledeen, K., and Lewis, H. R. Blown to Bits: your life, liberty, and happiness after the digital explosion. Upper Saddle River, N.J.: Addison-Wesley, 2008.

Ancillary Resources

Approved Programming Resources:

courses.projectstem.org/courses/14512

http://www.w3schools.com/js/default.asp (fadelk.com)
JS 101 (teaching-materials.org)
Learn JavaScript - Full 134-Part Course for Beginners (freecodecamp.org)
Hour of Code JavaScript Lesson Plan (kidscodemarin.com)
JavaScript Course - Free lessons (coursesweb.net)

Alice - This 3-D modeling environment allows students to create and animate 3-D worlds. This environment lends itself well to creating stories and games.

App Inventor - This open-source Web application allows students to create their own applications on mobile devices. App Lab - This is a programming environment for creating web applications with JavaScript. It allows students to develop programs and toggle back and forth between block-based and text-based programming modes.
EarSketch - This browser-based application allows students to create their own music using either JavaScript or Python. Greenfoot - This Java IDE is designed for use in education to create two-dimensional graphic applications, such as simulations and interactive games.
Java - There are several IDEs that can be used to write in Java. The Java language allows students to create and solve problems that vary widely in difficulty.
JavaScript - This language is commonly used to create interactive effects within Web browsers.
Lego Mindstorms NXT - This product integrates programming with Lego bricks and sensors to create and program robots. The instructions are assembled by linking together function blocks.
Processing - This programming language was initially created to serve as a software sketchbook, and it can be used to teach programming using a visual context.
Python - This language has the benefit of readability that might be helpful to new programmers.
Scratch - This blocks-based programming language allows students to build scripts to run animations. This product can be downloaded and installed on a computer or run in the browser.
Snap! - This Scratch-style programming language is block-based and allows users to define new primitives in JavaScript. Users can read and write information from the Internet using server-defined APIs and make mobile applications.
Swift - This programming language is designed for use with iOS, OS X, tvOS and watchOS. This environment allows students to create their own Apple apps and includes interactive environments that allow students to see the effects of changes or additions to code as they type.

Design and Development Process:

What Is the Software Development Life Cycle?" Official Blog Airbrake Bug Tracker. https://airbrake.io/blog/insight/what-is-the-software-development-life-cycle

"Engineering Design Process." https://www.teachengineering.org/ engrdesignprocess.php
"The Engineering Design Process." http://www.eie.org/overview/engineeringdesign-process Mohammed, Nabil, Ali Munassar, and A. Govardhan.
"A Comparison Between Five Models of Software Engineering." IJCSI International Journal of Computer Science 7.5 (2010): 94-101.

Open Source:

"What Is Open Source?" Opensource.com. https://opensource.com/resources/whatopen-source

Open Source Initiative. http://opensource.org/

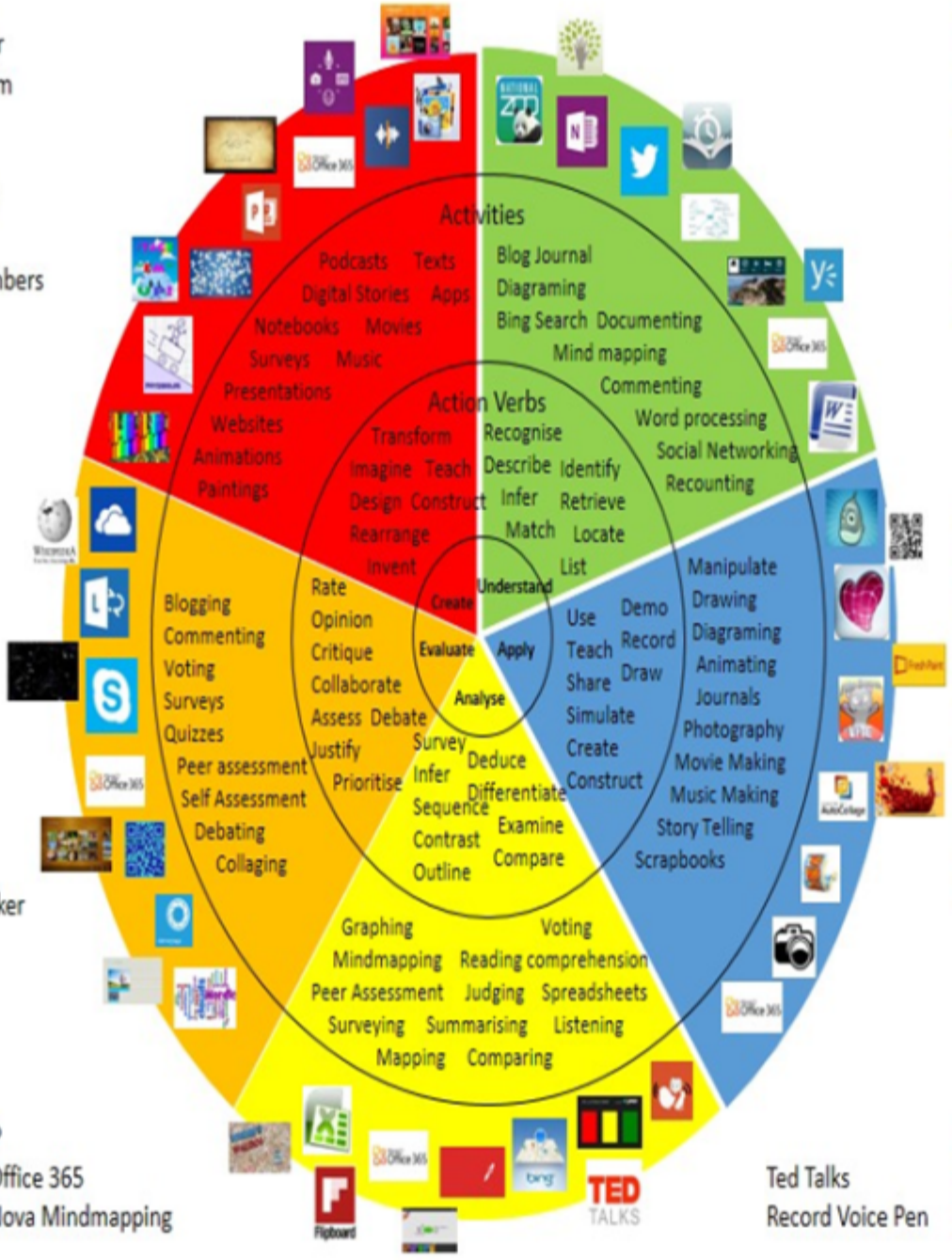
Technology Infusion

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

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.

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

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.

- 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

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

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

CRP.K-12.CRP1.1	Career-ready individuals understand the obligations and responsibilities of being a member of a community, and they demonstrate this understanding every day through their interactions with others. They are conscientious of the impacts of their decisions on others and the environment around them. They think about the near-term and long-term consequences of their actions and seek to act in ways that contribute to the betterment of their teams, families, community and workplace. They are reliable and consistent in going beyond the minimum expectation and in participating in activities that serve the greater good.
CRP.K-12.CRP2.1	Career-ready individuals readily access and use the knowledge and skills acquired through experience and education to be more productive. They make connections between abstract concepts with real-world applications, and they make correct insights about when it is appropriate to apply the use of an academic skill in a workplace situation.
CRP.K-12.CRP4.1	Career-ready individuals communicate thoughts, ideas, and action plans with clarity, whether using written, verbal, and/or visual methods. They communicate in the workplace with clarity and purpose to make maximum use of their own and others' time. They are excellent writers; they master conventions, word choice, and organization, and use effective tone and presentation skills to articulate ideas. They are skilled at interacting with others; they are active listeners and speak clearly and with purpose. Career-ready individuals think about the audience for their communication and prepare accordingly to ensure the desired outcome.
CRP.K-12.CRP7.1	Career-ready individuals are discerning in accepting and using new information to make decisions, change practices or inform strategies. They use reliable research process to search for new information. They evaluate the validity of sources when considering the use and adoption of external information or practices in their workplace situation.
CRP.K-12.CRP8.1	Career-ready individuals readily recognize problems in the workplace, understand the nature of the problem, and devise effective plans to solve the problem. They are aware of problems when they occur and take action quickly to address the problem; they thoughtfully investigate the root cause of the problem prior to introducing solutions. They carefully consider the options to solve the problem. Once a solution is agreed upon, they follow through to ensure the problem is solved, whether through their own actions or the actions of others.

Differentiation

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

One technique I use is when I teach a new Computer Science skill is using 'task cards' . For example, if we are doing a lesson on loops and there is clearly a pair of students in the class that has mastered the skill, I then have task cards with challenges ready on different skills or a harder looping concept.

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)

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

Adapting existing materials, simplifying or supplementing materials for Special Education Learning other options are below.

- **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.

- 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)

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

Advance Notes

One way that we can make things easier is by preparing and distributing advance notes. This gives ELLs the opportunity to preview what will be taught and, in turn, aids in comprehension of the material.

- 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

Intervention Strategies that will be employed in the unit

Parental Involvement

Planned intervention means involving parents. Do you have an agenda in place that goes home each night? Are parents also signing the agenda or contracts you have set up? How are you involving parental support at home for homework or additional follow up?

- 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)

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.

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

Javascript- Computer Science Course

Introduction to JavaScript

10.1: Intro to JS

Materials: Presentation (Included), Syntax Scavenger Hunt (Included), Syntax Scavenger Hunt Key (Included), Simple Bunny Game Code Handout (Included), Simple Bunny Game (Click “View Code” on the top right to remix)

1. Walk In: Get your colored pencils and fill out the top of your Syntax Scavenger Hunt handout.
2. Go over vocab from top of Syntax Scavenger Hunt
3. JavaScript Presentation
4. Start Scavenger Hunt using code handout
   **Note: I tell the students not to worry about getting everything right, but to see how much they can figure out. Some classes will need some of the next class to finish or to go over it
5. Show them the game

10.2: Explaining JavaScript

Materials: Explaining Bunny Code (Included), Explaining Bunny Code Key (Included), Simple Bunny Game (Click "View Code" on the top right to remix)

1. Walk In: Finish Scavenger Hunt
2. Go over scavenger hunt
   Note: I don't re-annotate the code as a class. For each section, I go over the answers to the questions and explain one or two examples in the code.
3. Share the Bunny Game code with the students, show them how to remix it (Click "View Code" on the top right) and how to comment out lines of code in case it helps them figure out what certain sections of code do.
   (**Note: only a few students actually use this technique, so if you have a distractible group, you might want to have them explain the code before they access it. Otherwise some of them get tempted to start modding the code before they have explained how it works)
4. Students work on explaining code and get it checked before they can move on

10.4: Modding

Materials: Bunny Mod Challenges (Included), Bunny Mod Challenges Key (Included)

1. Walk In: Pair up and finish explaining your code
2. What does modding mean? Modding is short for modifying code. It is popular in the video gaming world.

3. Students get their code explanations checked, then start working on the Bunny Mod Challenges. They should explain how they achieved each mod. When they finish, they can come up with their own extensions.

10.5: Continue Codding

10.6: Finish Modding

1. Walk In: What can you do in JavaScript that you couldn't in Scratch?
2. Share out
3. Finish modding
4. Gallery Walk

Note: At this point, if I have time, I often give students a week or so to work through some of Code.org's CS Discoveries curriculum. Since they have already learned the computational concepts, I make a packet of the concept overview pages from the CS Discoveries curriculum so they can read through and record how to do some of the things they learned in Scratch in JavaScript. Then I give them certain lessons to complete on Code.org to get them ready for the video game project. I have included my handout that guides them through sections of the CS Discoveries curriculum, but I have not included the concept overview pages as they are not mine to share.