Unit 3: Data Representation

Content Area: **Technology**
Course(s): **Sample Course**
Time Period: **NovDec**
Length: **Full year, Grades 9-12**
Status: **Published**

Data Representation, Computer Science

Department of Curriculum and Instruction

Belleville Public Schools

Curriculum Guide

Computer Science Principles, Grades 9-12

Data Representation

Belleville Board of Education

56 Ralph Street

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

Mr.Joseph Lepo, Director of Secondary Education

Board Approved:

Unit Overview

Explore the different means of representing information digitally.

Enduring Understanding

Multiple levels of abstraction are used to write programs or create other computational artifacts.

- Models and simulations use abstraction to generate new understanding and knowledge.
- There are trade-offs when representing information as digital data.
- Digital data representations involve trade-offs related to storage, security, and privacy concerns.
- Programming uses mathematical and logical concepts

Essential Questions

- How can computing extend traditional forms of human expression and experience?
- How are vastly different kinds of data, physical phenomena, and mathematical concepts represented on

a computer?
- How can computation be employed to help people process data and information to gain insight and knowledge?
- How can computation be employed to facilitate exploration and discovery when working with data?

Exit Skills

Students will be able to:

- Examine how numerical values are represented using different bases, including decimal and binary.
- Perform calculations for converting values from decimal to binary and binary to decimal as well as methods of counting in binary.
- Examine the exponential relationship between the number of digits and their range of representable values.
- Investigate how alphanumeric characters and symbols may be represented using ASCII and Unicode character mappings.
- Compare and contrast the implications of variable-width encodings (e.g., Morse code) vs. fixed-width encodings (e.g., Baudot code).
- Explore how the interpretation of binary data is dependent upon its intended format and use.
- Explore ways in which natural phenomena may be represented digitally.
- Analyze the extent to which digital approximations accurately reflect the reality that they represent.
- Compare and contrast discrete (digital) and continuous (analog) representations of natural phenomena.
- Examine the social implications of the ease with which perfect digital copies can be made.
- Examine the use of lists as ordered data structures that may contain multiple values.
- Implement the use of index values to represent the position of an item in a list.
- Analyze the implications of accessing an index position beyond the bounds of a list as well as the implications of case-sensitivity on ordered lists of strings.
- Investigate common operations for processing elements of a list, including searching for an element, removing an element, swapping the positions of two elements, or sorting an entire list into ascending or descending order.

New Jersey Student Learning Standards (NJSLS-S)

CS.9-12.8.1.12.CS.1	Describe ways in which integrated systems hide underlying implementation details to simplify user experiences.
CS.9-12.8.1.12.CS.2	Model interactions between application software, system software, and hardware.
CS.9-12.8.1.12.CS.3	Compare the functions of application software, system software, and hardware.
CS.9-12.CS	Computing Systems
TECH.9.4.12.CI	Creativity and Innovation
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.2	Identify career pathways that highlight personal talents, skills, and abilities (e.g., 1.4.12prof.CR2b, 2.2.12.LF.8).
TECH.9.4.12.CI.3	Investigate new challenges and opportunities for personal growth, advancement, and transition (e.g., 2.1.12.PGD.1).
TECH.9.4.12.CT.1	Identify problem-solving strategies used in the development of an innovative product or practice (e.g., 1.1.12acc.C1b, 2.2.12.PF.3).
TECH.9.4.12.CT.2	Explain the potential benefits of collaborating to enhance critical thinking and problem solving (e.g., 1.3E.12profCR3.a).
TECH.9.4.12.CT.3	Enlist input from a variety of stakeholders (e.g., community members, experts in the field) to design a service learning activity that addresses a local or global issue (e.g., environmental justice).
TECH.9.4.12.CT.4	Participate in online strategy and planning sessions for course-based, school-based, or other project and determine the strategies that contribute to effective outcomes.
	The usability, dependability, security, and accessibility of devices within integrated systems are important considerations in their design as they evolve.
	With a growth mindset, failure is an important part of success.
	Innovative ideas or innovation can lead to career opportunities.
	A computing system involves interaction among the user, hardware, application software, and system software.

Interdisciplinary Connections

LA.RL.11-12	Reading Literature
LA.RL.11-12.1	Cite strong and thorough textual evidence and make relevant connections to support analysis of what the text says explicitly as well as inferences drawn from the text, including determining where the text leaves matters uncertain.
LA.RL.11-12.2	Determine two or more themes or central ideas of a text and analyze their development over the course of the text, including how they interact and build on one another to produce a complex account; provide an objective summary of the text.
LA.RL.11-12.3	Analyze the impact of the author's choices regarding how to develop and relate elements of a story or drama (e.g., where a story is set, how the action is ordered, how the characters are introduced and developed).
LA.RL.11-12.4	Determine the meaning of words and phrases as they are used in the text, including figurative and connotative meanings; analyze the impact of specific word choices on meaning and tone, including words with multiple meanings or language that is particularly fresh, engaging, or beautiful. (e.g., Shakespeare as well as other authors.)
LA.L.11-12.4	Determine or clarify the meaning of unknown and multiple-meaning words and phrases based on grades 11–12 reading and content, choosing flexibly from a range of strategies.
LA.L.11-12.4.A	Use context (e.g., the overall meaning of a sentence, paragraph, or text; a word's position or function in a sentence) as a clue to the meaning of a word or phrase.
LA.L.11-12.4.B	Identify and correctly use patterns of word changes that indicate different meanings or parts of speech (e.g., conceive, conception, conceivable).
LA.L.11-12.4.C	Consult general and specialized reference materials (e.g., dictionaries, glossaries, thesauruses), both print and digital, to find the pronunciation of a word or determine or clarify its precise meaning, its part of speech, its etymology, or its standard usage.
LA.L.11-12.4.D	Verify the preliminary determination of the meaning of a word or phrase (e.g., by checking

the inferred meaning in context or in a dictionary).

Learning Objectives

Binary Encoding of Information:

1. Examine how numerical values are represented using different bases, including decimal and binary.
2. Perform calculations for converting values from decimal to binary and binary to decimal as well as methods of counting in binary.
3. Examine the exponential relationship between the number of digits and their range of representable values.
4. Investigate how alphanumeric characters and symbols may be represented using ASCII and Unicode character mappings.
5. Compare and contrast the implications of variable-width encodings (e.g., Morse code) vs. fixed-width encodings (e.g., Baudot code).
6. Explore how the interpretation of binary data is dependent upon its intended format and use.

Digital Approximations:

1. Explore ways in which natural phenomena may be represented digitally.
2. Analyze the extent to which digital approximations accurately reflect the reality that they represent.
3. Compare and contrast discrete (digital) and continuous (analog) representations of natural phenomena.
4. Examine the social implications of the ease with which perfect digital copies can be made.

Lists:

1. Examine the use of lists as ordered data structures that may contain multiple values.
2. Implement the use of index values to represent the position of an item in a list.
3. Analyze the implications of accessing an index position beyond the bounds of a list as well as the implications of case-sensitivity on ordered lists of strings.
4. Investigate common operations for processing elements of a list, including searching for an element, removing an element, swapping the positions of two elements, or sorting an entire list into ascending or descending order.

Suggested Activities & Best Practices

Case Studies

Refer to AP Classroom Resources for Computer Science Test

Use online modules through Code. Org, Edhesive and BJC, Snap.

Assignment Ideas/Suggested Activities

- **CS Unplugged**
- **CS Fundamentals Unplugged** from Code.org
- **CS4K12** Collection of handouts, worksheets and other documents to help teach Computer Science and Engineering concepts. Materials primarily target middle-school students (6th - 8th grade), but most can be used without modification for younger or older students.
- **Code.org YouTube Channel** > Learn about computer science
- **Nifty Assignments** Collection of fun, inspirational, or thought-provoking assignments for CS1 and CS2 shared at the annual SIGCSE conference
- **EngageCSEdu** Collection of activities and programming assignments designed to engage ALL students in CS1- and CS2-level courses. Materials have been reviewed by computer science educators and experts in student engagement and pedagogy.
- **Websheets** A system for creating and solving programming assignments **About Websheets**
- **Stanford CS Education Library** Education CS material from Stanford CS classes
- **Learn Java in N Games** Collection of game-based activities (including POGIL-style activities) for learning about Java programming. Can be used individually or as a complete curriculum. CS2 level.

Best Practices

- **CS Teaching Tips** Tips to help teachers anticipate students' difficulties and build upon students' strengths. Funded by a National Science Foundation Grant.
- Student will be able to identify famous African Americans in the field of computer programming
- Students will also be introduced to the basic energy efficient models in the programming field to help reduce global warming

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 to represent data within the unit.

Summative Assessments

• Chapter Test

• End of Chapter Projects from book.

Alternate Assessment

* Written reports

Benchmark Assessment

* Use online modules through Code. Org, Edhesive and BJC, Snap

- 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

Edhesive Online Curriculum, Code.Org

Ancillary Resources

General Resources:

- 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.

AP Approved Programming Resources:

(may choose one or more)

- 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.

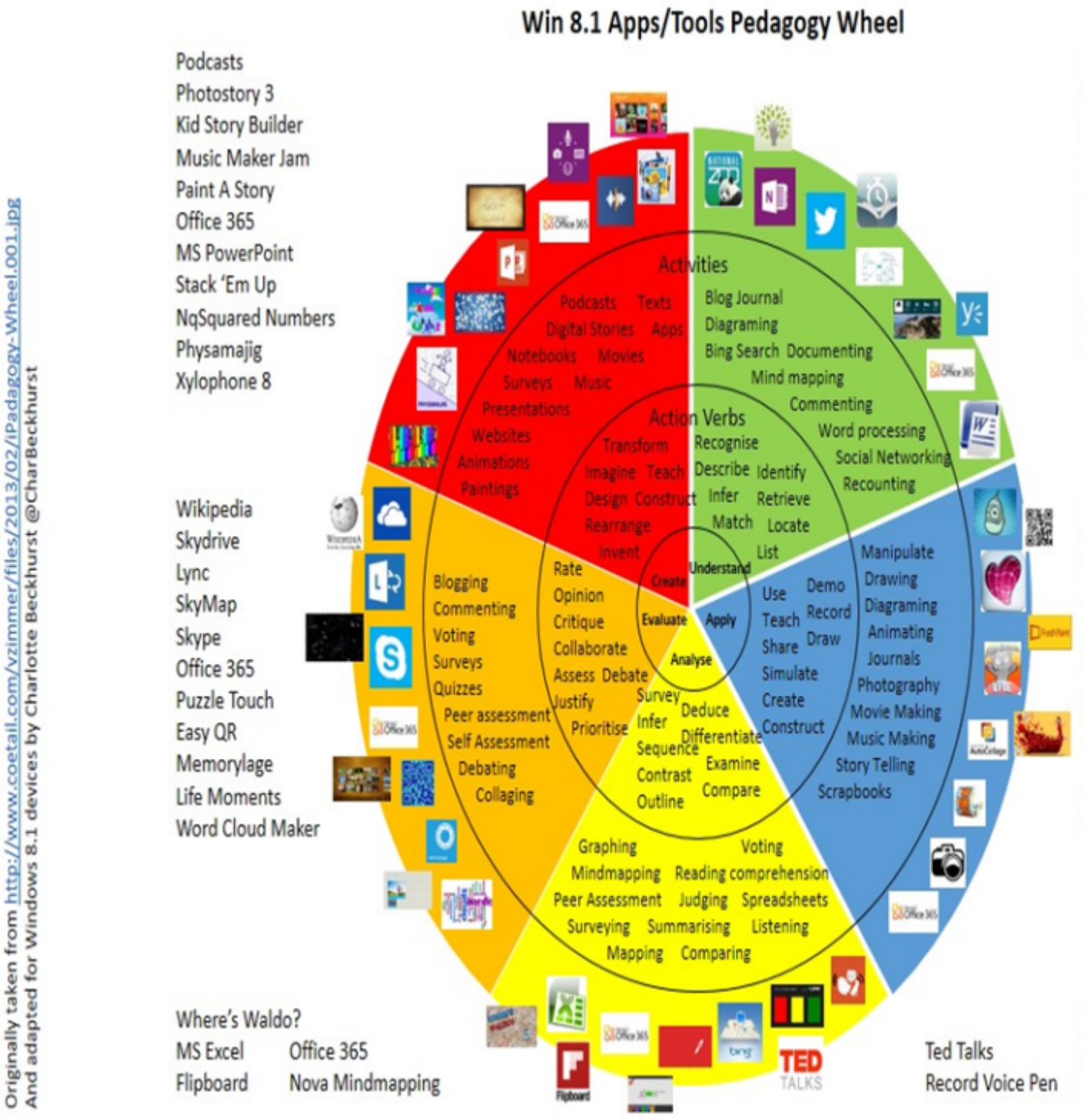
Open Source:

- “What Is Open Source?” Opensource.com. https://opensource.com/resources/whatopen-source
- Open Source Initiative. http://opensource.org/

Technology Infusion

Please reference video links and websites listed under Ancillary Resources and Suggested Activities & Best Practices.

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



Alignment to 21st Century Skills & Technology

WRK.9.2.12.CAP	Career Awareness and Planning
WRK.9.2.12.CAP.1	Analyze unemployment rates for workers with different levels of education and how the economic, social, and political conditions of a time period are affected by a recession.
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.3	Investigate how continuing education contributes to one's career and personal growth.
WRK.9.2.12.CAP.4	Evaluate different careers and develop various plans (e.g., costs of public, private, training schools) and timetables for achieving them, including educational/training requirements, costs, loans, and debt repayment.
WRK.9.2.12.CAP.5	Assess and modify a personal plan to support current interests and post-secondary plans.
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	Creativity and Innovation
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.2	Identify career pathways that highlight personal talents, skills, and abilities (e.g., 1.4.12prof.CR2b, 2.2.12.LF.8).
TECH.9.4.12.CI.3	Investigate new challenges and opportunities for personal growth, advancement, and transition (e.g., 2.1.12.PGD.1).
TECH.9.4.12.CT	Critical Thinking and Problem-solving
TECH.9.4.12.CT.1	Identify problem-solving strategies used in the development of an innovative product or practice (e.g., 1.1.12acc.C1b, 2.2.12.PF.3).
TECH.9.4.12.CT.2	Explain the potential benefits of collaborating to enhance critical thinking and problem solving (e.g., 1.3E.12profCR3.a).
TECH.9.4.12.CT.3	Enlist input from a variety of stakeholders (e.g., community members, experts in the field) to design a service learning activity that addresses a local or global issue (e.g., environmental justice).
	Career planning requires purposeful planning based on research, self-knowledge, and informed choices.
	With a growth mindset, failure is an important part of success.
	Innovative ideas or innovation can lead to career opportunities.
	There are strategies to improve one’s professional value and marketability.
	Collaboration with individuals with diverse experiences can aid in the problem-solving

process, particularly for global issues where diverse solutions are needed.

21st Century Skills/Interdisciplinary Themes

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

LA.RH.6-8.7	Integrate visual information (e.g., in charts, graphs, photographs, videos, or maps) with other information in print and digital texts. 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. Functions In school mathematics, functions usually have numerical inputs and outputs and are often defined by an algebraic expression. For example, the time in hours it takes for a car to drive 100 miles is a function of the car's speed in miles per hour, v; the rule T(v) = 100/v expresses this relationship algebraically and defines a function whose name is T.

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

• **Exemplar** - Technology Resources will be tailored for better student understanding of course material and

objectives to understand abstraction through models and visuals.

Additionally, 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;

Helping students understand that some learners need to move around to learn, while others do better sitting quietly (Tomlinson, 1995, 1999; Winebrenner, 1992, 1996).

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: 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

- 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: Extended Time

It's obvious that response time for ELLs is significantly greater than it is for students proficient in English. Given this, we know that ELLs may require more time to process and communicate information on assessments. To support your students in this area, give them additional time on tests to help.

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

Exemplar: Seating

Where are your students at risk? Hopefully, they are near a helping peer or with quick access to the teacher. Those with hearing or sight issues need to be close to the instruction which often means near the front.

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)

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

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://drive.google.com/file/d/108vVg35joLZBrnBUTG1OEuiUcd0bIept/view?usp=sharing

Lesson Plan – Bits & Bytes - Data

Teacher:	Corey Woodring.	Time Frame:	15 days
Grade:	9-12	School:	Belleville High School
Subject:	AP Computer Science Principles		

AP Essential Knowledge

(Referenced from CollegeBoard AP CS P Course & Exam Description)

DAT-1 The way a computer represents data internally is different from the way the data are interpreted and displayed for the user. Programs are used to translate data into a representation more easily understood by people.

A. Explain how data can be represented using bits.

1. Data values can be stored in variables, lists of items, or standalone constants and can be passed as input to (or output from) procedures.
2. Computing devices represent data digitally, meaning that the lowest-level components of any value are bits.
3. Bit is shorthand for binary digit and is either 0 or 1.
4. A byte is 8 bits.
5. Abstraction is the process of reducing complexity by focusing on the main idea. By hiding details irrelevant to the question at hand and bringing together related and useful details, abstraction reduces complexity and allows one to focus on the idea.
6. Bits are grouped to represent abstractions. These abstractions include, but are not limited to, numbers, characters, and color.
7. The same sequence of bits may represent different types of data in different contexts.
8. Analog data have values that change smoothly, rather than in discrete intervals, over time. Some examples of analog data include pitch and volume of music, colors of a painting, or position of a sprinter during a race.
9. The use of digital data to approximate real-world analog data is an example of abstraction.
10. Analog data can be closely approximated digitally using a sampling technique, which means measuring values of the analog signal at regular intervals called samples. The samples are measured to figure out the exact bits required to store each sample.

A. Explain the consequences of using bits to represent data.

1. In many programming languages, integers are represented by a fixed number of bits, which limits the range of integer values and mathematical operations on those values. This limitation can result in overflow or other errors.

2. Other programming languages provide an abstraction through which the size of representable integers is limited only by the size of the computer's memory; this is the case for the language defined in the exam reference sheet.

3. In programming languages, the fixed number of bits used to represent real numbers limits the range and mathematical operations on these values; this limitation can result in roundoff and other errors. Some real numbers are represented as approximations in computer storage.

- EXCLUSION STATEMENT: Specific range limitations for real numbers are outside the scope of this course and the AP Exam.

A. For binary numbers: a. Calculate the binary (base 2) equivalent of a positive integer (base 10)

and vice versa.

b. Compare and order binary numbers.

1. Number bases, including binary and decimal, are used to represent data.

2. Binary (base 2) uses only combinations of the digits zero and one.

3. Decimal (base 10) uses only combinations of the digits 0 – 9.

4. As with decimal, a digit's position in the binary sequence determines its numeric value. The numeric value is equal to the bit's value (0 or 1) multiplied by the place value of its position.

5. The place value of each position is determined by the base raised to the power of the position. Positions are numbered starting at the rightmost position with 0 and increasing by 1 for each subsequent position to the left.

CRD – 1

A. Explain how computing innovations are improved through collaboration

    1. A computing innovation includes a program as an integral part of its function.

    2. A computing innovation can be physical (e.g. self-driving car), nonphysical computing software (e.g. picture editing software), or a nonphysical computing concept (e,g. e-commerce).

    3. Effective collaboration produces a computing innovation that reflects the diversity of talents and perspectives of those who designed it.

    4. Collaboration that includes diverse perspectives helps avoid bias in the development of computing innovations.

    5. Consultation and communication with users are important aspects of the development computing innovations.

    6. Information gathered from potential users can be used to understand the purpose of a program from diverse perspectives and to develop a program that fully incorporates these perspectives.

B. Explain how computing innovations are developed by groups of people.

    1. Online tools support collaboration by allowing programmers to share and provide feedback on

ideas and documents.

2. Common models such as pair programming exist to facilitate collaboration.

C. Demonstrate effective interpersonal skills during collaboration.

1. Effective collaborative teams practice interpersonal skills, including gbut not limited to:

i. Communication

ii. Consensus building

iii. Conflict resolution

iv. negotiation

CRD – 2

A. Explain how a program or code segment functions.

1. A program is a collection of program statements that performs a specific task when run by a computer. A program is often referred to as software.

2. A code segment is a collection of program statements that is part of a program.

3. A program needs to work for a variety of inputs and situations.

4. The behavior of a program is how a program functions during execution and is often described by how a user interacts with it.

5. A program can be described broadly by what it does or in more detail by both what the program does and how the program statements accomplish this function.

B. Identify input(s) to a program

1. Program inputs are data sent to a computer for processing by a program. Input can come in a variety of forms, such as tactile, audio, visual, or text

2. An event is associated with an action and supplies input data to a program.

3. Events can be generated when a key is pressed, a mouse is clicked, a program is started, or any other defined action occurs that affects the flow of execution.

4. Inputs usually affect the output produced by a program.

5. In event-driven programming, program statements are executed when triggered rathe rthan through the sequential flow of control.

6. Input can come from a user or other programs.

C. Identify output(s) produced by a program.

1. Program outputs are any data sent from a program to a device. Program output can come in a variety of forms, such as tactile, audio, visual, or text.

2. Program output is usually based on a program's input or prior state (e.g. internal values).

D. Develop a program using a development process.

1. A development process can be ordered and intentional, or exploratory in nature.
2. There are multiple development processes. The following phases are commonly used when developing a program:
    i. Investigating and reflecting
    ii. Designing
    iii. Prototyping
    iv. Testing
3. A development process that is iterative requires refinement and revision based on feedback, testing, or reflection throughout the process. This may require revisiting earlier phases of the process.
4. A development process that is incremental is one that breaks the problem into smaller pieces and makes sure each piece works before adding it to the whole.

E. Design a program and its user interface.

1. The design of a program incorporates investigation to determine its requirements.
2. Investigation in a development process is useful for understanding and identifying the program constraints, as well as the concerns and interests of the people who will use the program.
3. Some ways investigation can be performed are as follows:
    i. Collecting data through surveys
    ii. User testing
    iii. Interviews
    iv. Direct observations
4. Program requirements describe how a program functions and may include a description of user interactions that a program must provide.
5. A program's specification defines the requirements for the program.
6. In a development process, the design phase outlines how to accomplish a given program specification.
7. The design phase of a program may include:
    i. Brainstorming
    ii. Planning and storyboarding

iii. Organizing the program into modules and functional components

iv. Creation of diagrams that represent the layouts of the user interface

v. Development of a testing strategy for the program

I. For errors in an algorithm or program: Identify the error and correct the error.

1. A logic error is a mistake in the algorithm or program that causes it to behave incorrectly or unexpectedly.

2. A syntax error is a mistake in the program where the rules of the programming language are not followed.

3. A run-time error is a mistake in the program that occurs during the execution of a program. Programming languages define their own run-time errors.

4. An overflow error is an error that occurs when a computer attempts to handle a number that is outside of the defined range of values.

5. The following are effective ways to find and correct errors:

    i. Test cases

    ii. Hand tracing

    iii. Visualizations

    iv. Debuggers

    v. Adding extra output statement(s)

J. Identify inputs and corresponding expected outputs or behaviors that can be used to check the correctness of an algorithm or program

1. In the development process, testing uses defined inputs to ensure tha tan algorithm or program is producing the expected outcomes. Programmers use the results from testing to revise their algorithms or programs.

2. Defined inputs used to test a program should demonstrate the different expected outcomes that are at or just beyond the extremes (minimum and maximum) of input data.

3. Program requirements are needed to identify appropriate defined inputs for testing.

Enduring Understanding & CTP Skills

(Referenced from CollegeBoard AP CS P Course & Exam Description)

1. Computational Solution Design

    A. Investigate the situation, context or task.

5. Computing Innovations

A. Explain how computing systems work.

6. Responsible Computing

A. Collaborate in the development of solutions.

Essential Questions

(Some referenced from CollegeBoard AP CS P Course & Exam Description)

(What questions will the student be able to answer as a result of the instruction?)

1. How can we understand the abstraction levels of software and hardware?
2. How does a decimal number system work?
3. How do we convert between number systems?
4. How do we convert from number systems to ASCII?

Assessment

(What is acceptable evidence to show desired results (rubrics, exam, etc.)? Attach Copy

- Complete various written checkpoint exercises that focus on the explanation and description of computer programming, pseudocode, and python.
- 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.
- Unit 3 Assessment

Formative Evaluations:

Formative Assessment with polling

Classwork/Homework

Summative Evaluations:

Unit 7 Test/ReTest

Quizzes

AP Classroom Big Idea 2 Formative Topic Questions (see Sequence and Scope for when to assign problems)

Lab work

Sequence and Scope

Day	Topic/Activities	CW-HW
1	• Computers & Abstraction	Problems 1-10
2	• Number Systems	Problems 11-18
3	• Base-2 (Binary) Number System	Problems 19-22
4	• Base-2 (Binary) Number System	Problems 23-28
5	• Quiz 1	
6	• Base-8 & Base-16 Number Systems	Problems 29-34
7	• Base-8 & Base-16 Number Systems	Problems 35-40
8	• Base-8 & Base-16 Number Systems	Problems 41-44, AP Topic Questions 2.1
9	• Bits & Bytes	Problems 45-50
10	• Bits & Bytes	Problems 51-58, AP Topic Questions 2.2 – 2.4
11	• Quiz 2	
12-14	• Lab	Study for Test
15	• Unit 7 Assessment	None

CS.9-12.8.1.12.CS.1	Describe ways in which integrated systems hide underlying implementation details to simplify user experiences.
CS.9-12.8.1.12.CS.2	Model interactions between application software, system software, and hardware.
CS.9-12.8.1.12.CS.3	Compare the functions of application software, system software, and hardware.
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.CS	Computing Systems
TECH.9.4.12.CI	Creativity and Innovation

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.2	Identify career pathways that highlight personal talents, skills, and abilities (e.g., 1.4.12prof.CR2b, 2.2.12.LF.8).
TECH.9.4.12.CI.3	Investigate new challenges and opportunities for personal growth, advancement, and transition (e.g., 2.1.12.PGD.1).
TECH.9.4.12.CT	Critical Thinking and Problem-solving
TECH.9.4.12.CT.1	Identify problem-solving strategies used in the development of an innovative product or practice (e.g., 1.1.12acc.C1b, 2.2.12.PF.3).
	A computing system involves interaction among the user, hardware, application software, and system software.
	With a growth mindset, failure is an important part of success.
	The usability, dependability, security, and accessibility of devices within integrated systems are important considerations in their design as they evolve.
	Innovative ideas or innovation can lead to career opportunities.
	Successful troubleshooting of complex problems involves multiple approaches including research, analysis, reflection, interaction with peers, and drawing on past experiences.
	Collaboration with individuals with diverse experiences can aid in the problem-solving process, particularly for global issues where diverse solutions are needed.