

Unit 4: Digital Media Processing

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

Computer Science Principles

Department of Curriculum and Instruction



Belleville Public Schools

Curriculum Guide

AP Computer Science Principles, Grades 9-12

Digital Media Processing

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

Use of a coding environment to programmatically manipulate digital images and audio.

Digitalization, is the process of converting information into a digital (i.e. computer-readable) format, in which the information is organized into bits. The result is the representation of an object, image, sound, document or signal (usually an analog signal) by generating a series of numbers that describe a discrete set of its points or samples. The result is called digital representation or, more specifically, a digital image, for the object, and digital form, for the signal. In modern practice, the digitized data is in the form of binary numbers, which facilitate computer processing and other operations, but, strictly speaking, digitizing simply means the conversion of analog source material into a numerical format; the decimal or any other number system that can be used instead.

Enduring Understanding

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

Enduring understandings:

- Summarize important ideas and core processes that are central to a discipline and have lasting value beyond the classroom;
- Synthesize what students should understand - not just know or do - as a result of studying a particular content area;
- Frame the Big Ideas that give meaning and lasting importance to such discrete curriculum elements as facts and skills;
- Transfer to other fields and adult life;
- "Unpack" areas of the curriculum where students may struggle to gain understanding or where they demonstrate misunderstandings and misconceptions;
- Provide a conceptual foundation for studying the content area;
- Articulate what students should "revisit" over the course of their lifetimes in relationship to the content area;
- Are framed as declarative sentences that present major curriculum generalizations and recurrent ideas.

Examples:

- **Enduring Understanding:** Reading is a process by which we construct meaning about the information being communicated by an author within a print or non-print medium.
- **Essential Question:** How is reading a process for constructing meaning from text?

Essential Questions

How does abstraction help us in writing programs, creating computational artifacts, and solving problems?

- How can computational models and simulations help generate new understanding and knowledge?

- Why are some languages better than others when used to implement algorithms?

Exit Skills

- Students will be able to use appropriate terminology when describing the size of digital files.
- Identify and compare the size of familiar digital media.
- Explain why the optimal amount of compression is impossible or “hard” to identify.
- Explain factors that make compression challenging.
- Describe the purpose and rationale for lossless compression.
- Explain how images are encoded with pixel data.
- Describe a pixel as an element of a digital image.
- Encode a B&W image in binary representing both the pixel data (intensity) and metadata (width, height).
- Explain why image width and height are metadata for a digital image.
- Use the Pixelation Tool to encode small color images with varying bits-per-pixel settings.
- Explain the color encoding scheme for digital images.
- Determine the benefits of using hexadecimal numbers for representing long streams of bits
- Explain the difference between lossy and lossless compression.
- Explain the relative benefits or drawbacks of different file formats, particularly in terms of how they compress information.
- Explain the difference between open source and licensed software.

New Jersey Student Learning Standards (NJSL-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.9-12.8.1.12.NI.1	Evaluate the scalability and reliability of networks, by describing the relationship between routers, switches, servers, topology, and addressing.

CS.9-12.CS	Computing Systems
CS.9-12.NI	Networks and the Internet
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).
	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.
	Collaboration with individuals with diverse experiences can aid in the problem-solving process, particularly for global issues where diverse solutions are needed.
	Successful troubleshooting of complex problems involves multiple approaches including research, analysis, reflection, interaction with peers, and drawing on past experiences.
	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.

Interdisciplinary Connections

LA.RL.11-12	Reading Literature
	Key Ideas and Details
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.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.

Learning Objectives

Procedural Programming:

- Compare and contrast the programming capabilities of a visual programming language with those of a text-based programming language.
- Write programs in a text-based programming language that make use of parameterized methods to invoke specific behaviors.
- Understand the importance of using proper punctuation and syntax when coding in a text-based programming language.
- Use event handlers to respond to mouse and keyboard input.
- Write code using common programming constructs like conditional statements for selection and loops for iteration.

Image Manipulation:

- Examine the structure of images as compositions of pixels.
- Explore methods of representing color, including the various colors that can be produced by the combination of different ratios of red, green, and blue light.
- Modify the color channels of pixels in an image to produce a variety of effects.
- Design algorithms for modifying the pixels in an image in prescribed ways to create custom image filters.
- Explore encoding schemes of common image file formats.

Audio Manipulation:

- Analyze the differences between analog and digital sound.
- Investigate the roles that sampling rate and bit depth play in determining the quality of digitized sound.
- Explore methods of programmatically generating digital audio as well as altering and modifying digital audio by adjusting volume, pitch, and sampling rate.
- Evaluate the effects of compression algorithms in reducing the amount of data needed to represent an audio sample.

Global Impact:

- Evaluate the positive and negative consequences of digitally altering images.
- Discuss the ethics of digitally manipulating images.
- Discuss the issues related to intellectual property.

- Explore the limitations and rights associated with a number of common licenses, including Creative Commons.

Suggested Activities & Best Practices

In this project based learning unit students will learn on the Edhesive simulator that:

Students will use various problem solving strategies to collect input, store information, and generate outputs.

Students will learn to read and write information to data files.

This unit also investigates various ethical issues arising in computer science.

Students will also explore and learn how to create arrays, and how to write programs that declare, initialize, modify and access these arrays.

Students will write algorithms with nested structures, and sub-programs, and algorithms that perform simple data management tasks.

- 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

Best Practices

- [CS Teaching Tips](#) Tips to help teachers anticipate students' difficulties and build upon students' strengths. Funded by a National Science Foundation Grant.

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

Written reports

Benchmark Assessments

Students will write algorithms with nested structures, and sub-programs, and algorithms that perform simple data management tasks.

- Admit Tickets
- Anticipation Guide
- Common Benchmarks
- Compare & Contrast
- Create a Multimedia Poster
- DBQ's
- Define
- Describe

- Evaluate
- Evaluation rubrics
- Exit Tickets
- Explaining
- Fist- to-Five or Thumb-Ometer
- Illustration
- Journals
- KWL Chart
- Learning Center Activities
- Multimedia Reports
- Newspaper Headline
- Outline
- Question Stems
- Quickwrite
- Quizzes
- Red Light, Green Light
- Self- assessments
- Socratic Seminar
- Study Guide
- Surveys
- Teacher Observation Checklist
- Think, Pair, Share
- Think, Write, Pair, Share
- Top 10 List
- Unit review/Test prep
- Unit tests
- Web-Based Assessments
- Written Reports

Primary Resources & Materials

Code.org curriculum, Edhesive curriculum

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](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/whatoopen-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

Win 8.1 Apps/Tools Pedagogy Wheel

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

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

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

Ted Talks
 Record Voice Pen



Originally taken from <http://www.coetail.com/vzimmer/files/2013/02/1Padagogy-Wheel.001.jpg>
 And adapted for Windows 8.1 devices by Charlotte Beckhurst @CharBeckhurst

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.
WRK.9.2.12.CAP.7	Use online resources to examine licensing, certification, and credentialing requirements at the local, state, and national levels to maintain compliance with industry requirements in areas of career interest.
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.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. Innovative ideas or innovation can lead to career opportunities. There are strategies to improve one's professional value and marketability. With a growth mindset, failure is an important part of success. Career planning requires purposeful planning based on research, self-knowledge, and informed choices.

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.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.
Functions

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.CRP11.1 Career-ready individuals find and maximize the productive value of existing and new technology to accomplish workplace tasks and solve workplace problems. They are flexible and adaptive in acquiring new technology. They are proficient with ubiquitous technology applications. They understand the inherent risks-personal and organizational-of technology applications, and they take actions to prevent or mitigate these risks.

CRP.K-12.CRP12.1 Career-ready individuals positively contribute to every team, whether formal or informal. They apply an awareness of cultural difference to avoid barriers to productive and positive interaction. They find ways to increase the engagement and contribution of all team members. They plan and facilitate effective team meetings.

Differentiation

Exemplar: Flipped Classroom, Video to support class lesson

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

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)

Supporting lesson with additional resources such as instructional videos, you tube and online instruction where applicable

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

- **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)

Exemplar: eDictionaries / Google translate

When used correctly, [technology never fails to transform a classroom](#). Allow your students to use eDictionaries in your classroom to look up unknown words. Any regular dictionary will do, but eDictionaries allow students to hear the pronunciations of words. As you might imagine, this is extremely helpful for anyone learning a second language.

English Language Learning adaptations that can 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 clarify
- allowing products (projects, timelines, demonstrations, models, drawings, dioramas, poster boards, charts, graphs, slide shows, videos, etc.) to demonstrate student's learning;
- allowing students to correct errors (looking for understanding)

- allowing the use of note cards or open-book during testing
- decreasing the amount of work presented or required
- having peers take notes or providing a copy of the teacher's notes
- modifying tests to reflect selected objectives
- providing study guides
- reducing or omitting lengthy outside reading assignments
- reducing the number of answer choices on a multiple choice test
- tutoring by peers
- using computer word processing spell check and grammar check features
- using true/false, matching, or fill in the blank tests in lieu of essay tests

At Risk

Exemplar: Peer Support

Sometimes, all you have to do is [assign a peer](#) to help keep a student at risk on task. Peers can help build confidence in other students by assisting in peer learning.

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

Talented and Gifted Learning (T&G)

Exemplar: Grouping • Group gifted students with other gifted students or higher-level learners.

Talented and Gifted adaptations that can be employed in this 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

Lesson Plan – Graphics & Images

Teacher:	Corey Woodring	Time Frame:	14 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. Compare data compression algorithms to determine which is best in a particular context.
 1. Data compression can reduce the size (number of bits) of transmitted or stored data.
 2. Fewer bits does not necessarily mean less information.
 3. The amount of size reduction from compression depends on both the amount of redundancy in the original data representation and the compression algorithm applied.
 4. Lossless data compression algorithms can usually reduce the number of bits stored or transmitted while

guaranteeing complete reconstruction of the original data.

5. Lossy data compression algorithms can significantly reduce the number of bits stored or transmitted but only allow reconstruction of an approximation of the original data.
6. Lossy data compression algorithms can usually reduce the number of bits stored or transmitted more than lossless compression algorithms.
7. In situations where quality or ability to reconstruct the original is maximally important, lossless compression algorithms are typically chosen.
8. In situations where minimizing data size or transmission time is maximally important, lossy compression algorithms are typically chosen.

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

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 rather than 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 that an 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.
 - b. Determine and design an appropriate method or approach to achieve the purpose.
 - c. Explain how collaboration affects the development of a solution.
 - d. Evaluate solution options.

1. Abstraction in Program Development
 - a. Generalize data sources through variables.
 - b. Use abstraction to manage complexity in a program.
 - c. Explain how abstraction manages complexity.

1. Computing Innovations

- a. Explain how computing systems work.
- b. Explain how knowledge can be generated from data.
- c. Describe the impact of a computing innovation.
- d. Describe the impact of gathering data.
- e. Evaluate the use of computing based on legal and ethical factors.

2. Responsible Computing

- a. Collaborate in the development of solutions.
- b. Use safe and secure methods when using computing devices.
- c. Acknowledge the intellectual property of others.

1. Essential Questions

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

1.

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

1. How can images and text be compressed to save storage space?
2. How are images and text encoded?
3. Why is text and image compression necessary?

1. 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

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	• Encoding Digital Images	Problems 1-6
2	• Encoding Digital Images	Problems 7-12
3	• Encoding Digital Images	Problems 13-16
4	• Text Compression	Problems 17-24
5	• Text Compression	Problems 25-28
6	• Text Compression	Problems 29-33
7	• Quiz 1	
8	• Compressing File Formats	Problems 34-39
9	• Compressing File Formats	Problems 40-44, AP Classroom Topic Questions 2.2
10-13	• Lab	Study for Test
14	• Unit 8 Assessment	None

Summative Evaluations:

Unit 8 Test/ReTest

CS.9-12.8.1.12.CS.2	Model interactions between 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	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).
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.
TECH.9.4.12.GCA.1	<p>Collaborate with individuals to analyze a variety of potential solutions to climate change effects and determine why some solutions (e.g., political, economic, cultural) may work better than others (e.g., SL.11-12.1., HS-ETS1-1, HS-ETS1-2, HS-ETS1-4, 6.3.12.GeoGI.1, 7.1.IH.IPERS.6, 7.1.IL.IPERS.7, 8.2.12.ETW.3).</p> <p>Collaboration with individuals with diverse experiences can aid in the problem-solving process, particularly for global issues where diverse solutions are needed.</p> <p>A computing system involves interaction among the user, hardware, application software, and system software.</p> <p>The usability, dependability, security, and accessibility of devices within integrated systems are important considerations in their design as they evolve.</p> <p>Innovative ideas or innovation can lead to career opportunities.</p> <p>With a growth mindset, failure is an important part of success.</p>