Honors Computer Science Principles

Grades 11-12

Unit 1

Subject	Grades	Unit	Suggested Timeline
Honors Computer Science Principles	11-12	1 – Algorithms, Graphics, and Graphical User Interfaces	17 weeks

Grade Level Summary

Computer Science Principles (CSP) is a specialization course in Project Lead The Way's (PLTW) high school engineering pathway. Students practice problem solving beginning with structured activities and progressing to open-ended projects and problems that require them to develop planning, documentation, communication, and other professional skills. The mastery of a single programming language is not the goal of CSP. Instead, the course aims to develop computational thinking, generate excitement about the field of computing, and introduce computational tools that foster creativity. The course also aims to build students' awareness of the tremendous demand for computer specialists and as well as professionals with computational skills in all fields. Each unit focuses on one or more computationally intensive career paths. The course also aims to engage students to consider issues raised by the present and future societal impact of computing. CSP is aligned with the College Board's AP Computer Science Principles curriculum.

Grade Level Units

Unit 1 – Algorithms, Graphics, and Graphical User Interfaces
Unit 2 – The Internet
Unit 3 – Raining Reigning Data
Unit 4 – Intelligent Behavior

Unit Title

Algorithms, Graphics, and Graphical User Interfaces

Unit Overview

The goal of Unit 1 is to excite students about programming and to build their algorithmic thinking and ability to use abstraction. Student creativity is emphasized as they work with programming languages such as Scratch, App Inventor, and Python to tell graphical stories, publish games, Android applications, and explore various development environments and programming techniques. Students will create original code and read and modify code provided from other sources. An Agile software development process is emphasized and personal, professional, and collaborative skills are refined. Students will also consider policy questions about the ownership and control of digital data and examine the implications for creative industries and consumers. In this unit students begin their exploration of career paths tied to computing.

Unit Essential Questions

1. How do computers perform complicated tasks built from simple instructions?
2. How are variables used in programming?
3. How do programmers approach a complicated problem?
4. What role does creativity play in algorithmic programming?
5. What makes for a good process for collaborative software development?
6. What do programming languages and development environments have in common?
7. What can be represented by binary data?
8. What contributes to an effective process for software development?
9. How can a program be analyzed, understood, and modified?
10. How does abstraction make the software development process easier?

Key Understandings

1. Computing fosters creative expression, sometimes resulting in artifacts
2. Computational artifacts can be evaluated
3. Programming is a creative endeavor
4. Binary sequences represent digital data
5. Computing relies on layers of abstraction in software
6. Abstraction allows for simple utilization of other people's code
7. Solutions to complex problems can be encapsulated in reusable components
8. The solution to one problem can be applied to another seemingly unrelated problem by identifying and reusing a pattern
9. Physical systems, like sound or biological molecules, have both digital and analog characteristics
10. Programs implement algorithms to solve problems

11. What are the practices that lead to effective collaboration? 12. How is computing affecting the way we live our lives? 13. How will computing change our world?	11. Algorithms can be analyzed for efficiency, and appropriate algorithms can be selected based upon efficiency 12. Empirical analysis of algorithms requires a systematic approach 13. A given algorithmic problem with standard solutions can be applied in diverse contexts 14. Creating solutions with computation requires exploring the tools available, selecting an appropriate tool, and gaining expertise with the tool 15. Solutions in a programming language are created by breaking a problem apart into component problems 16. Creating solutions with computation requires a persistent, iterative problem-solving approach 17. Programming requires an understanding of mathematical operations and data abstractions 18. The user interface of a piece of software can greatly affect how it is used 19. Functions with arguments make code modular and reusable 20. Programmers create high-level documentation to communicate the purpose and function of their code. 21. Programmers must prioritize making their code well-documented and readable for it to be maintained 22. Mobile and networked computing have transformed commerce, social interactions, news sourcing and dissemination, and culture 23. Assistive technologies using hardware and software can extend human capabilities 24. Making information accessible to all people requires attention from a variety of stakeholders 25. New opportunities for human creativity and innovation exist because of networked, mobile, and embedded computing 26. Computing artifacts and programs can be higher quality as a result of collaboration 27. Working in a team requires effective communication, clear responsibilities, and attention to interpersonal relationships 28. Collaboration allows communities to create software that can impact people's lives 29. Creative ideas and technical solutions must be communicated in a clear and concise manner 30. How people present themselves affects how their work is received 31. Computer science and information technology careers offer creative job opportunities for individuals with a wide variety of backgrounds and goals 32. Parallel computing is a quickly evolving field relevant to hardware, software, and users 33. Computational thinking boosts most career paths

Focus Standards Addressed in the Unit	
3.4.10.B4	Recognize that technological development has been evolutionary, the result of a series of refinements to a basic invention.
15.4.12.B	Evaluate the impact of social, legal, ethical, and safe behaviors on digital citizenship.

15.4.12.H	Use programming languages to develop logical thinking and problem solving skills.
15.4.12.J	Create a complex computer program to solve a problem.
CC.1.3.11-12.J	Acquire and use accurately general academic and domain-specific words and phrases, sufficient for reading, writing, speaking, and listening at the college and career readiness level; demonstrate independence in gathering vocabulary knowledge when considering a word or phrase important to comprehension or expression.
CC.1.4.11-12.F	Demonstrate a grade-appropriate command of the conventions of standard English grammar, usage, capitalization, punctuation, and spelling.
CC.2.1.HS.F.3	Apply quantitative reasoning to choose and interpret units and scales in formulas, graphs and data displays.
CC.2.3.HS.A.14	Apply geometric concepts to model and solve real world problems.
CC.2.4.HS.B.7	Apply the rules of probability to compute probabilities of compound events in a uniform probability model**.**
CC.3.5.11-12.C	Follow precisely a complex multistep procedure when carrying out experiments, taking measurements, or performing technical tasks; analyze the specific results based on explanations in the text.
CC.3.5.11-12.D	Determine the meaning of symbols, key terms, and other domain-specific words and phrases as they are used in a specific scientific or technical context relevant to grades 11–12 texts and topics .
CC.3.5.11-12.G	Integrate and evaluate multiple sources of information presented in diverse formats and media (e.g., quantitative data, video, multimedia) in order to address a question or solve a problem.
CC.3.5.11-12.I	Synthesize information from a range of sources (e.g., texts, experiments, simulations) into a coherent understanding of a process, phenomenon, or concept, resolving conflicting information when possible.
CC.3.6.11-12.B	Write informative/explanatory texts, including the narration of historical events, scientific procedures/experiments, or technical processes.
CC.3.6.11-12.C	Produce clear and coherent writing in which the development, organization, and style are appropriate to task, purpose, and audience.
CC.3.6.11-12.E	Use technology, including the Internet, to produce, publish, and update individual or shared writing products in response to ongoing feedback, including new arguments or information.
CC.3.6.11-12.F.	Conduct short as well as more sustained research projects to answer a question (including a self-generated question) or solve a problem; narrow or broaden the inquiry when appropriate; synthesize multiple sources on the subject, demonstrating understanding of the subject under investigation.
CC.3.6.11-12.G	Gather relevant information from multiple authoritative print and digital sources, using advanced searches effectively; assess the strengths and limitations of each source in terms of the specific task, purpose, and audience; integrate information into the text selectively to maintain the flow of ideas, avoiding plagiarism and overreliance on any one source and following a standard format for citation.
CC.3.6.11-12.H	Draw evidence from informational texts to support analysis, reflection, and research.
CC.3.6.11-12.I	Write routinely over extended time frames (time for reflection and revision) and shorter time frames (a single sitting or a day or two) for a range of discipline-specific tasks, purposes, and audiences.

Important Standards Addressed in the Unit	
15.4.12.C	Develop criteria for analyzing hardware options to meet defined needs.
CC.2.3.HS.A.1	Use geometric figures and their properties to represent transformations in the plane.

CC.3.5.11-12.B	Determine the central ideas or conclusions of a text; summarize complex concepts, processes, or information presented in a text by paraphrasing them in simpler but still accurate terms.
CC.3.6.11-12.A	Write arguments focused on discipline-specific content.

Misconceptions	**Proper Conceptions**
1. It is important to be focused on one specific programming language to solve problems.	1. Good computer programming begins with computational thinking. An abstract understanding of the use of elements such as variables and objects will allow a programmer to adapt to new languages more quickly.
2. A good program consists on nothing but good code.	2. Programmers need to fully document their work so that modifications and enhancements can quickly be made. As time passes, it becomes more difficult to jump back into your own program and it is even more difficult for a new programmer to make changes to a poorly documented program.
3. A functional user interface is more important than its visual appearance.	3. A good GUI requires a balance between function and visual appeal. Creativity – in both layout and programming – is a valuable skill in a computing field.

Concepts	**Competencies**	**Vocabulary**
• Algorithms	• Create a program with a big plan using several small modules	Absolute filename Abstraction Accessibility Accumulator Variable Aggregator Variable Agile Design Algorithm Algorithmic Problem Algorithmic Solution Alpha channel Analog Analog-to-Digital Conversion API Arguments Array ASCII Assignment Operator Attribute Avatar Base Case Base n Best-so-far Variable Binary Bit Block of Code Boolean Expression Bounding Box Bug Built-in Function Button Byte Call Catch Central Angle Character Class Class Diagrams
• Mobile App Development	• Design, document, manage, troubleshoot, and present an application that fills a need for users.	
• Object Manipulation	• Manipulate images through iteration and object-oriented methods. Navigate a filesystem tree and examine filenames and metadata.	
• GUI Creation	• Create a user interface using code. Analyze and suggest, justify, and make improvements to an existing user interface.	

Vocabulary (cont'd)

Clone
Code
Collection
Compiler
Compound Conditional
Compression
Compression Ratio
Concatenation
Conditional
Constructor
Continuous
Data Abstraction
Debugging
Default Value
Design Patterns
Diff
Digit
Digital
Discrete
Docstring
Element
Encapsulation
Ergonomics
Escape sequences /escape character
Evaluate
Event
Event Handler
Exception
Execute
Fidelity
Fixed Variable
Float
Flow Chart
Formal Arguments
Function Name
Garbage Collection
Glass Box Testing
Global Scope
Global Variable
GUI
Handler
Hexadecimal
Human-Computer Interaction
If-structure
Immutable
Import
Input
Instance
Instantiation
Integrated Development Environment (IDE)
Interpolate
Interpreter
Iterable
Iteration
Keyword-value pair
Kilobyte
Least Significant BitsLibrary
Local Scope
Loop
Lossy
Machine Code
Megabyte
Memory
Metadata
Meta-information
Method
Method Call
Model-View-Controller
Modular Code
Module
Most Recent Variable
Multi-line Comment
Multiple Assignment
Mutable
Namespace
Native Type
Node
Null String
Object
Object-oriented
Observer
Octal
One-way Flag Variable
Op Code
Opaque
Output
Overriding a default
Padding
Palette
Pixel
Pixelate
Procedural Abstraction
Procedure Definition
Product Backlog
Programming Paradigm
Pseudocode
Pseudorandom
Recursion
Register
Relative filename
Repetitive Strain Injuries
Return String
Return value
State Root node
Scope
Script
Scrum
Searching a List
Sector
Slicing
Slider
Sorting a List
Sprint Task List
Sprite
Standard PositionState
State Diagram
Steganography
Stepper Variable
Syntax
Technical Debt
Test Suite
Test-Driven Design
Traceback
Tracer Route
Tree
Tuple
Turing Test
Type
Type Casting
Unified Modeling Language (UML)
Unique
User Stories
UTF-8
Validate
Value
Variable Binding
Variable Roles
Velocity
Version Control
Video Card
Virtual Reality (VR)
Walker Variable
Waterfall Design
Widget
Working Directory

Assessments

Engineering Journal Checks – Students will maintain a formal engineering journal to document their work throughout the course. Periodic checks will assess proper notebook format and content.

Oral Presentations – Students will periodically present their progress to their teacher and peers.

Unit Tests / Unit Projects – Each unit may include a summative written test or project. Projects may be assessed through a presentation, engineering notebook review, electronic submission, or a combination of one or more of these.

Course Portfolio – Students will compile a portfolio to document their body of work in the course

Suggested Strategies to Support Design of Coherent Instruction

Charlotte Danielson's Framework for Teaching: Domain 3 Instruction

Unit 1 sets the table for creative, computational, logical thinking throughout the course. Intentionally discuss how abstraction in programming allows us to transcend any specific language. While object oriented programming will be used, discuss how it relates and contrasts with functional programming.

Frequently allow students to share and comment on each other's programming work. The open-ended nature of the problems that will be solved in this course along with the multiple programming styles that will exist in a class section will inherently create many solutions allowing students to learn from each other.

Differentiation:

- Provide graphic organizers
- Provide multiple concrete examples
- Break extended projects into smaller identifiable milestones with checkpoints along the way
- Pair stronger students with struggling students for peer assistance

Interdisciplinary Connections:

- Design process – Scientific method
- Research process – English / Social Studies
- Writing skills – English
- Graphics - Art

Additional Resources:

- ***https://csta.acm.org/Curriculum/sub/K12Standards.html*** Computer Science Teachers Association K-12 standards
- ***https://code.org/*** An open resource for additional programming references
- ***https://www.computer.org/*** Institue of Electrical and Electronics Engineers professional computer society website

Created By:

Rick Geesaman

Honors Computer Science Principles

Grades 11-12

Unit 2

Subject	Grades	Unit	Suggested Timeline
Honors Computer Science Principles	11-12	2 – The Internet	8 weeks

Grade Level Summary
Computer Science Principles (CSP) is a specialization course in Project Lead The Way's (PLTW) high school engineering pathway. Students practice problem solving beginning with structured activities and progressing to open-ended projects and problems that require them to develop planning, documentation, communication, and other professional skills. The mastery of a single programming language is not the goal of CSP. Instead, the course aims to develop computational thinking, generate excitement about the field of computing, and introduce computational tools that foster creativity. The course also aims to build students' awareness of the tremendous demand for computer specialists and as well as professionals with computational skills in all fields. Each unit focuses on one or more computationally intensive career paths. The course also aims to engage students to consider issues raised by the present and future societal impact of computing. CSP is aligned with the College Board's AP Computer Science Principles curriculum.

Grade Level Units
Unit 1 – Algorithms, Graphics, and Graphical User Interfaces
Unit 2 – The Internet
Unit 3 – Raining Reigning Data
Unit 4 – Intelligent Behavior

Unit Title
The Internet

Unit Overview
The goal of Unit 2 is for students to have a more concrete understanding of the Internet as a set of computers exchanging bits and the implications of these exchanges. Students will use PHP and SQL to structure and access a database hosted on a remote server, learn how HTML and CSS direct the client computer to render a page, and experiment with JavaScript to provide dynamic content. The focus of the unit is on the protocols that allow the Internet to function securely to deliver social media and e-commerce content. Students will work in several Web languages to understand how the languages work together to deliver this content. The history and workings of the Internet are explored, and issues of security, privacy, and democracy are considered. Practical cyber security hygiene is included. Career paths in cyber security, web development, and information technology are highlighted.

Unit Essential Questions
1. How does the Internet work?
2. How can we protect ourselves, our privacy, and our assets when working on the Internet?
3. How has the Internet affected society?
4. What is the nature of attack and defense in cybersecurity?

Key Understandings
1. Computing fosters creative expression, sometimes resulting in artifacts
2. Computational artifacts can be evaluated
3. Programming is a creative endeavor
4. Binary sequences represent digital data
5. Computing relies on layers of abstraction in software
6. Computing relies on abstractions of hardware represented with software
7. The solution to one problem can be applied to another seemingly unrelated problem by identifying and reusing a pattern
8. Data can be structured to facilitate use
9. The size of a data set affects how the data can be used
10. Collecting and managing data raises technical issues regarding storage, access, durability, privacy, and security
11. Ethical and societal issues are raised by the impact of Big Data and require attention from many stakeholders

12. Programs implement algorithms to solve problems
13. Algorithms can be analyzed for efficiency, and appropriate algorithms can be selected based upon efficiency
14. Empirical analysis of algorithms requires a systematic approach
15. A given algorithmic problem with standard solutions can be applied in diverse contexts problems
16. Creating solutions with computation requires exploring the tools available, selecting an appropriate tool, and gaining expertise with the tool
17. Solutions in a programming language are created by breaking a problem apart into component problems
18. Creating solutions with computation requires a persistent, iterative problem-solving approach
19. Programming requires an understanding of mathematical operations and data abstractions
20. The user interface of a piece of software can greatly affect how it is used
21. Functions with arguments make code modular and reusable
22. Programmers create high-level documentation to communicate the purpose and function of their code
23. Programmers must prioritize making their code well-documented and readable for it to be maintained
24. Networked and mobile computing rely on various protocols to provide services
25. A variety of languages are used for Web programming, with both overlapping and complementary purposes
26. The Internet facilitates collaboration
27. Device-to-device communication through the Internet passes through a non-unique route
28. The information and processing power on any networked device can be accessed by potentially hostile parties
29. Maintaining a safe presence on the Internet requires attention and knowledge
30. Cybersecurity depends on hardware and software components, including cryptography
31. Mobile and networked computing have transformed commerce, social interactions, news sourcing and dissemination, and culture
32. Computing is having profound impacts on individual privacy
33. Making information accessible to all people requires attention from a variety of stakeholders
34. New opportunities for human creativity and innovation exist because of networked, mobile, and embedded computing
35. Networked infrastructure affects and is affected by commercial and governmental structures and policies
36. Scalability is an important consideration for distributed solutions
37. Computing artifacts and programs can be higher quality as a result of collaboration
38. Working in a team requires effective communication, clear responsibilities, and attention to interpersonal relationships
39. Collaboration allows communities to create software that can impact people's lives

	40. Creative ideas and technical solutions must be communicated in a clear and concise manner 41. How people present themselves affects how their work is received 42. Computer science and information technology careers offer creative job opportunities for individuals with a wide variety of backgrounds and goals 43. Parallel computing is a quickly evolving field relevant to hardware, software, and users 44. Computational thinking boosts most career paths

Focus Standards Addressed in the Unit	
3.4.10.B4	Recognize that technological development has been evolutionary, the result of a series of refinements to a basic invention.
15.4.12.B	Evaluate the impact of social, legal, ethical, and safe behaviors on digital citizenship.
15.4.12.F	Compare and contrast network environments, including the function of network devices and connectivity issues.
15.4.12.K	Evaluate advanced multimedia work products and make recommendations based on the evaluation.
15.4.12.H.	Use programming languages to develop logical thinking and problem solving skills.
CC.1.3.11-12.J	Acquire and use accurately general academic and domain-specific words and phrases, sufficient for reading, writing, speaking, and listening at the college and career readiness level; demonstrate independence in gathering vocabulary knowledge when considering a word or phrase important to comprehension or expression.
CC.1.4.11-12.F	Demonstrate a grade-appropriate command of the conventions of standard English grammar, usage, capitalization, punctuation, and spelling.
CC.2.4.HS.B.2	Summarize, represent, and interpret data on two categorical and quantitative variables.
CC.2.4.HS.B.3	Analyze linear models to make interpretations based on the data.
CC.3.5.11-12.C	Follow precisely a complex multistep procedure when carrying out experiments, taking measurements, or performing technical tasks; analyze the specific results based on explanations in the text.
CC.3.5.11-12.D	Determine the meaning of symbols, key terms, and other domain-specific words and phrases as they are used in a specific scientific or technical context relevant to grades 11–12 texts and topics .
CC.3.5.11-12.G	Integrate and evaluate multiple sources of information presented in diverse formats and media (e.g., quantitative data, video, multimedia) in order to address a question or solve a problem.
CC.3.5.11-12.I	Synthesize information from a range of sources (e.g., texts, experiments, simulations) into a coherent understanding of a process, phenomenon, or concept, resolving conflicting information when possible.
CC.3.6.11-12.C	Produce clear and coherent writing in which the development, organization, and style are appropriate to task, purpose, and audience.
CC.3.6.11-12.E	Use technology, including the Internet, to produce, publish, and update individual or shared writing products in response to ongoing feedback, including new arguments or information.
CC.3.6.11-12.F	Conduct short as well as more sustained research projects to answer a question (including a self-generated question) or solve a problem; narrow or broaden the inquiry when appropriate; synthesize multiple sources on the subject, demonstrating understanding of the subject under investigation.
CC.3.6.11-12.G	Gather relevant information from multiple authoritative print and digital sources, using advanced searches effectively; assess the strengths and limitations of each source in terms of the specific task, purpose, and audience; integrate information into the text selectively to maintain the flow of

	ideas, avoiding plagiarism and overreliance on any one source and following a standard format for citation.
CC.3.6.11-12.H	Draw evidence from informational texts to support analysis, reflection, and research.
CC.3.6.11-12.I	Write routinely over extended time frames (time for reflection and revision) and shorter time frames (a single sitting or a day or two) for a range of discipline-specific tasks, purposes, and audiences.

Important Standards Addressed in the Unit	
CC.2.2.HS.C.2	Graph and analyze functions and use their properties to make connections between the different representations.
CC.2.2.HS.C.5	Construct and compare linear, quadratic and exponential models to solve problems.
CC.3.5.11-12.B	Determine the central ideas or conclusions of a text; summarize complex concepts, processes, or information presented in a text by paraphrasing them in simpler but still accurate terms.
CC.3.6.11-12.A	Write arguments focused on discipline-specific content.
CC.3.6.11-12.B	Write informative/explanatory texts, including the narration of historical events, scientific procedures/experiments, or technical processes.

Misconceptions	**Proper Conceptions**
1. If it is on the internet, it is free for all to use.	1. Just because something can be accessed on the internet does not negate the fact that someone owns it. Some entities make their material freely available without restriction but other objects require permission or payment to use.
2. There is nothing worth taking on my computer so I am safe surfing the internet.	2. While stored data is a potentially valuable item, the exchange of data between computers leaves all users potentially vulnerable when using the internet.

Concepts	**Competencies**	**Vocabulary**
• The Internet • Shopping / Social Networking • Security	• Describe the protocols that result in a rendered page when a URL is entered into a web browser • Create a web page and analyze code to discover and remove inefficiencies • Describe how networked computing affects governmental entities	Absolute Path Addon Adware Antivirus Software Association for Computing Machinery (ACM) Authenticate Authoritative DNS AWSS Bandwidth Big-O Notation Black-Hat Penetration Testing Blacklist and Whitelist Bookmark Botnet Browser Browser Tab Brute Force Brute Force Attack Certificate Authority Ciphertext Client Application Client Machine Client-Side Scripting

Vocabulary (cont'd)

Computable
Computer and Information Technology
Computer Emergency Readiness Team (CERT)
Computer System Design
Constant Function
Cookie
Credentials
Cryptography
CSS
Cyber Hygiene
DBMS
Decryption
Denial of Service (DoS) Attack
Derived Data Set
Distributed Denial of Service (DDoS) Attack
Domain
Domain Name System
Dotted Decimal Notation
e-commerce
Efficiency
Empirical Efficiency
Encryption
Enumeration
Escalation of Privileges
Ethernet
Exploit
Exponential Function
External Style Sheet
Filter
Firewall
First Normal Form
Flag
FTP
GET
Hacking
History
HTML
HTML5
HTTP
ICANN
IETF
Inline Style Internal Style Sheet
Internet
InversesIPv4
IPv6
ISP
JavaScript
Keystroke Logger
Kill
Latency
Linear Function
Malware
MySQL
Name Server
NIC
Noise
Normalization
NoSQL
Obscurity
One-Way Function
Open Data Movement
P and NP Problems
Packets
Paginating
Paired Keys
Parallel Computing
Parameters
Patch
Path
Payload
Penetration Testing
Permission Privileges
PHP
Plaintext
Polynomial Function
Polynomial Time
Port
POST
Primary Key
Process
Property-Value Pair
Protocol
Public Key Encryption
Quadratic Function Query
Recursive DNS
Redundancy
Relational Database
Relative Frequencies
Relative Path
Remote Shell
RFC
Rogue Security Software
Rootkit
Router
RSA Algorithm
Script Kiddie
Scrum Poker
Second Normal Form
Selector
Server-Side Script
Short Circuit
Social Engineering
Spam Filter
Spyware
SSH
SSL Certificate
Subdomain
Substitution Cipher
System Administrator
TCP Handshake
TCP/IP
Theoretical Efficiency
Third Normal Form
Third-Party Cookie
Time Complexity
Top-Level Domain
Tractable
Trojan Horse.
URL
User Account Control
Virus
Vulnerability
W3C Technical Report
W3C Working Group
Web Crawler
Web Index
Wildcard
Worm
Worst-Case Running Time
WYSIWIG

Assessments

Engineering Journal Checks – Students will maintain a formal engineering journal to document their work throughout the course. Periodic checks will assess proper notebook format and content.

Oral Presentations – Students will periodically present their progress to their teacher and peers.

Unit Tests / Unit Projects – Each unit may include a summative written test or project. Projects may be assessed through a presentation, engineering notebook review, electronic submission, or a combination of one or more of these.

Course Portfolio – Students will compile a portfolio to document their body of work in the course

Suggested Strategies to Support Design of Coherent Instruction

Charlotte Danielson's Framework for Teaching: Domain 3 Instruction

Rely on students to provide examples of good and bad websites and actively discuss what makes each site what it is, how it could be improved, and what structures are in place to make the sites render as they do. Researching existing sites will be a good inspiration for those who feel that they struggle creatively.

Differentiation:

- Provide graphic organizers
- Provide multiple concrete examples
- Break extended projects into smaller identifiable milestones with checkpoints along the way
- Pair stronger students with struggling students for peer assistance

Interdisciplinary Connections:

- Design process – Scientific method
- Research process – English / Social Studies
- Writing skills – English
- Graphics - Art

Additional Resources:

- ***https://csta.acm.org/Curriculum/sub/K12Standards.html*** Computer Science Teachers Association K-12 standards
- ***https://code.org/*** An open resource for additional programming references
- ***https://www.computer.org/*** Institue of Electrical and Electronics Engineers professional computer society website

Created By:

Rick Geesaman

Honors Computer Science Principles

Grades 11-12

Unit 3

Subject	Grades	Unit	Suggested Timeline
Honors Computer Science Principles	11-12	3 – Raining Reigning Data	7 weeks

Grade Level Summary

Computer Science Principles (CSP) is a specialization course in Project Lead The Way's (PLTW) high school engineering pathway. Students practice problem solving beginning with structured activities and progressing to open-ended projects and problems that require them to develop planning, documentation, communication, and other professional skills. The mastery of a single programming language is not the goal of CSP. Instead, the course aims to develop computational thinking, generate excitement about the field of computing, and introduce computational tools that foster creativity. The course also aims to build students' awareness of the tremendous demand for computer specialists and as well as professionals with computational skills in all fields. Each unit focuses on one or more computationally intensive career paths. The course also aims to engage students to consider issues raised by the present and future societal impact of computing. CSP is aligned with the College Board's AP Computer Science Principles curriculum.

Grade Level Units

Unit 1 – Algorithms, Graphics, and Graphical User Interfaces
Unit 2 – The Internet
Unit 3 – Raining Reigning Data
Unit 4 – Intelligent Behavior

Unit Title

Raining Reigning Data

Unit Overview

The goal of Unit 3 is for students to see the availability of large-scale data collection and analysis in every area they can imagine. Students examine very large data sets tied to themselves as well as to areas of work and society. They learn a variety of data visualization techniques and work to recognize opportunities to apply algorithmic thinking and automation when considering questions that have answers embedded in data. The complexity of the data sets, visualizations, and analysis increases in complexity as the unit progresses, challenging students to generalize concepts developed in earlier activities.

Unit Essential Questions

1. How will computation impact fields other than computing itself?
2. How will computation impact society?
3. How can patterns be discovered in data?
4. How has computation changed biology?

Key Understandings

1. Computing fosters creative expression, sometimes resulting in artifacts
2. Computational artifacts can be evaluated
3. Programming is a creative endeavor
4. Binary sequences represent digital data
5. Simulation and modeling can help us understand, communicate about, and predict natural phenomena
6. Data can be structured to facilitate use
7. Our capabilities to collect, store, and process data are changing at profound rates
8. Analysis of data can be automated
9. Data visualizations are important tools for discovering and communicating knowledge
10. The human brain and today's computers have complementary strengths for analyzing data
11. The size of a data set affects how the data can be used
12. Collecting and managing data raises technical issues regarding storage, access, durability, privacy, and security

13. Ethical and societal issues are raised by the impact of Big Data and require attention from many stakeholders
14. Programs implement algorithms to solve problems
15. Algorithms can be analyzed for efficiency, and appropriate algorithms can be selected based upon efficiency
16. Empirical analysis of algorithms requires a systematic approach
17. A given algorithmic problem with standard solutions can be applied in diverse contexts
18. Creating solutions with computation requires exploring the tools available, selecting an appropriate tool, and gaining expertise with the tool
19. Solutions in a programming language are created by breaking a problem apart into component problems
20. Creating solutions with computation requires a persistent, iterative problem-solving approach
21. Programming requires an understanding of mathematical operations and data abstractions
22. Functions with arguments make code modular and reusable
23. Programmers create high-level documentation to communicate the purpose and function of their code
24. Programmers must prioritize making their code well-documented and readable for it to be maintained
25. The Internet facilitates collaboration
26. The information and processing power on any networked device can be accessed by potentially hostile parties
27. Mobile and networked computing have transformed commerce, social interactions, news sourcing and dissemination, and culture
28. Computing is having profound impacts on individual privacy
29. Assistive technologies using hardware and software can extend human capabilities
30. Making information accessible to all people requires attention from a variety of stakeholders
31. New opportunities for human creativity and innovation exist because of networked, mobile, and embedded computing
32. Crowdsourcing identifies new problems and provides new solutions
33. Scalability is an important consideration for distributed solutions
34. Computing artifacts and programs can be higher quality as a result of collaboration
35. Working in a team requires effective communication, clear responsibilities, and attention to interpersonal relationships
36. Creative ideas and technical solutions must be communicated in a clear and concise manner
37. How people present themselves affects how their work is received
38. Computer science and information technology careers offer creative job opportunities for individuals with a wide variety of backgrounds and goals
39. Computational thinking boosts most career path

Focus Standards Addressed in the Unit	
3.4.10.B4	Recognize that technological development has been evolutionary, the result of a series of refinements to a basic invention.
15.4.12.B	Evaluate the impact of social, legal, ethical, and safe behaviors on digital citizenship.
15.4.12.F	Compare and contrast network environments, including the function of network devices and connectivity issues.
15.4.12.H	Use programming languages to develop logical thinking and problem solving skills.
CC.1.3.11-12.J	Acquire and use accurately general academic and domain-specific words and phrases, sufficient for reading, writing, speaking, and listening at the college and career readiness level; demonstrate independence in gathering vocabulary knowledge when considering a word or phrase important to comprehension or expression.
CC.1.4.11-12.F	Demonstrate a grade-appropriate command of the conventions of standard English grammar, usage, capitalization, punctuation, and spelling.
CC.2.2.HS.C.2	Graph and analyze functions and use their properties to make connections between the different representations.
CC.2.2.HS.C.5	Construct and compare linear, quadratic and exponential models to solve problems.
CC.2.4.HS.B.1	Summarize, represent, and interpret data on a single count or measurement variable.
CC.2.4.HS.B.2	Summarize, represent, and interpret data on two categorical and quantitative variables.
CC.2.4.HS.B.3	Analyze linear models to make interpretations based on the data.
CC.2.4.HS.B.4	Recognize and evaluate random processes underlying statistical experiments.
CC.2.4.HS.B.5	Make inferences and justify conclusions based on sample surveys, experiments, and observational studies.
CC.3.5.11-12.C	Follow precisely a complex multistep procedure when carrying out experiments, taking measurements, or performing technical tasks; analyze the specific results based on explanations in the text.
CC.3.5.11-12.D	Determine the meaning of symbols, key terms, and other domain-specific words and phrases as they are used in a specific scientific or technical context relevant to grades 11–12 texts and topics .
CC.3.5.11-12.G	Integrate and evaluate multiple sources of information presented in diverse formats and media (e.g., quantitative data, video, multimedia) in order to address a question or solve a problem.
CC.3.5.11-12.I	Synthesize information from a range of sources (e.g., texts, experiments, simulations) into a coherent understanding of a process, phenomenon, or concept, resolving conflicting information when possible.
CC.3.6.11-12.C	Produce clear and coherent writing in which the development, organization, and style are appropriate to task, purpose, and audience.
CC.3.6.11-12.E	Use technology, including the Internet, to produce, publish, and update individual or shared writing products in response to ongoing feedback, including new arguments or information.
CC.3.6.11-12.F	Conduct short as well as more sustained research projects to answer a question (including a self-generated question) or solve a problem; narrow or broaden the inquiry when appropriate; synthesize multiple sources on the subject, demonstrating understanding of the subject under investigation.
CC.3.6.11-12.G	Gather relevant information from multiple authoritative print and digital sources, using advanced searches effectively; assess the strengths and limitations of each source in terms of the specific task, purpose, and audience; integrate information into the text selectively to maintain the flow of ideas, avoiding plagiarism and overreliance on any one source and following a standard format for citation.

CC.3.6.11-12.H	Draw evidence from informational texts to support analysis, reflection, and research.
CC.3.6.11-12.I	Write routinely over extended time frames (time for reflection and revision) and shorter time frames (a single sitting or a day or two) for a range of discipline-specific tasks, purposes, and audiences.

Important Standards Addressed in the Unit	
CC.2.4.HS.B.6	Use the concepts of independence and conditional probability to interpret data.
CC.3.5.11-12.B	Determine the central ideas or conclusions of a text; summarize complex concepts, processes, or information presented in a text by paraphrasing them in simpler but still accurate terms.
CC.3.6.11-12.A	Write arguments focused on discipline-specific content.
CC.3.6.11-12.B	Write informative/explanatory texts, including the narration of historical events, scientific procedures/experiments, or technical processes.

Misconceptions	**Proper Conceptions**
1. The more data that is available, the better the decisions based on that data will be.	1. Unfortunately, too much data can be overwhelming and can lead to invalid decisions. Using algorithms to look for patterns, correlations, and erroneous data along with data visualization generally leads to more meaningful results than looking for causal relationships in a big data set.
2. Having data will eliminate uncertainty.	2. Especially when data includes human factors, uncertainty can never be eliminated. The goal though should be to analyze and use the data in a way that reduces uncertainty. Data cannot always be relied upon as the sole decision maker but it can always be used to inform decisions.

Concepts	**Competencies**	**Vocabulary**
• Visualizing Data • Discovering Knowledge from Data	• Clean, select, sort, and visualize a given set of data. Describe how the availability of digital data affects society. • Recognize and describe linear correlation (or lack of correlation) between two variables in set of data. Describe how a set of geographic data could be used to establish patterns.	Abstraction in Models Anomaly Bar Graph Big Data Box Categorical Variable Center of a Distribution Click-Through Rate Comma Separated Value File (CSV) Cones Confidence Interval Data Mining, KDD Data Parallel Data Skills Deidentified Data Derived Data Device Fingerprinting Dictionary Disaggregating Data Distributed Edge / Link Electronic Frontier Foundation End User License Agreement (EULA) Exploratory Data Analysis

Vocabulary (cont'd)

Fault-Tolerant	Monte Carlo Simulation	Sample Distribution
FrameFrequency	Narrow AI	Scatter Plot
Generalization	Neurons	Sensitive Information
Graph	Node	Shape of Distribution
Graphics Processing Unit (GPU)	Normal Distribution	Simulation
Hard-Coding	Occipital Lobe	Spread of a Distribution
Histogram	Opt-In, Opt-Out Clauses	Standard Deviation
Impression	Parallel Processing	Strong AI
Inferential Statistics	Parameter	Targeted Advertising
Inferential Statistics	Pie Chart	Task Parallel
In-Place	Population Distribution	Terms of Service
Interquartile Range	Privacy Policy	The Standard Normal Distribution
Intervals / Classes / Bins	p-valueQuantitative Variable	Threads
Latitude	Range	Training Set
Longitude	Redundant	Transformation
Mean	Reidentification	Uniform Distribution
Median	Relative Frequency	Validation Set
Mode	Relative Reference Retina	View-Through Rate
Model	Rods	Visualization

Assessments

Engineering Journal Checks – Students will maintain a formal engineering journal to document their work throughout the course. Periodic checks will assess proper notebook format and content.

Oral Presentations – Students will periodically present their progress to their teacher and peers.

Unit Tests / Unit Projects – Each unit may include a summative written test or project. Projects may be assessed through a presentation, engineering notebook review, electronic submission, or a combination of one or more of these.

Course Portfolio – Students will compile a portfolio to document their body of work in the course

Suggested Strategies to Support Design of Coherent Instruction

Charlotte Danielson's Framework for Teaching: Domain 3 Instruction

All students will have dealt with a data set at some point, if for nothing else to calculate measures of central tendency. For many students though, this will be their first exploration of how one set of data can provide varying degrees of information and patterns. Provide multiple examples of small data sets with small numbers of variables and how those sets can be visualized in different ways. Students should also be given the opportunity to infer the structure of a data set when given only visualization(s) of that data set.

Differentiation:

- Provide graphic organizers
- Provide multiple concrete examples
- Break extended projects into smaller identifiable milestones with checkpoints along the way
- Pair stronger students with struggling students for peer assistance

Interdisciplinary Connections:

- Design process – Scientific method
- Research process – English / Social Studies
- Writing skills – English
- Graphics - Art

Additional Resources:

- ***https://csta.acm.org/Curriculum/sub/K12Standards.html*** Computer Science Teachers Association K-12 standards
- ***https://code.org/*** An open resource for additional programming references
- ***https://www.computer.org/*** Institue of Electrical and Electronics Engineers professional computer society website

Created By:

Rick Geesaman

Honors Computer Science Principles

Grades 11-12

Unit 4

Subject	Grades	Unit	Suggested Timeline
Honors Computer Science Principles	11-12	4 – Intelligent Behavior	7 weeks

Grade Level Summary

Computer Science Principles (CSP) is a specialization course in Project Lead The Way's (PLTW) high school engineering pathway. Students practice problem solving beginning with structured activities and progressing to open-ended projects and problems that require them to develop planning, documentation, communication, and other professional skills. The mastery of a single programming language is not the goal of CSP. Instead, the course aims to develop computational thinking, generate excitement about the field of computing, and introduce computational tools that foster creativity. The course also aims to build students' awareness of the tremendous demand for computer specialists and as well as professionals with computational skills in all fields. Each unit focuses on one or more computationally intensive career paths. The course also aims to engage students to consider issues raised by the present and future societal impact of computing. CSP is aligned with the College Board's AP Computer Science Principles curriculum.

Grade Level Units

Unit 1 – Algorithms, Graphics, and Graphical User Interfaces
Unit 2 – The Internet
Unit 3 – Raining Reigning Data
Unit 4 – Intelligent Behavior

Unit Title

Intelligent Behavior

Unit Overview

In Unit 4 the emergence of intelligent behavior is explored from two distinct approaches: from human crowd sourcing of data and from separate algorithmic agents working in parallel. The goal is to galvanize the connections among computing concepts and between computing and society. The unit begins by exploring the hardware layer of computing, working from discrete components to integrated circuits. The exponential advancement of electronics, low on the ladder of abstraction, is connected to advancements at the highest levels on the ladder of abstraction, where artificial intelligence and simulation and modeling are impacting all fields. As the unit progresses, students identify problems and questions that can be addressed with computer simulation, incorporating agent-based modeling. Students are challenged to explore the assumptions and parameters built into several simulations and to attach meaning to the results. Having explored a few applications of intelligent behavior emerging from algorithmic components, students reflect on the current and future state of artificial intelligence.

Unit Essential Questions

1. How are simulations created from models?
2. How are simulation models similar to and different from reality?
3. How are modeling and simulation impacting other creative fields?
4. How has computation affected our ability to predict the future?
5. How has computation affected our ability to experience virtual phenomena?
6. How has simulation changed the design process in engineering and other creative fields?

Key Understandings

1. Computing fosters creative expression, sometimes resulting in artifacts
2. Computational artifacts can be evaluated
3. Programming is a creative endeavor
4. Binary sequences represent digital data
5. Computing relies on layers of abstraction in software
6. Computing relies on abstractions of hardware represented with software
7. Abstraction allows for simple utilization of other people's code
8. The solution to one problem can be applied to another seemingly unrelated problem by identifying and reusing a pattern
9. Simulation and modeling can help us understand, communicate about, and predict natural phenomena
10. Physical systems, like sound or biological molecules, have both digital and analog characteristics

	11. Intelligent behavior emerges from networked collections of simple algorithms 12. Data can be structured to facilitate use 13. Our capabilities to collect, store, and process data are changing at profound rates 14. Analysis of data can be automated 15. Data visualizations are important tools for discovering and communicating knowledge 16. The human brain and today's computers have complementary strengths for analyzing data 17. Programs implement algorithms to solve problems 18. Empirical analysis of algorithms requires a systematic approach 19. A given algorithmic problem with standard solutions can be applied in diverse contexts 20. Creating solutions with computation requires exploring the tools available, selecting an appropriate tool, and gaining expertise with the tool 21. Creating solutions with computation requires a persistent, iterative problem-solving approach 22. Programming requires an understanding of mathematical operations and data abstractions 23. The user interface of a piece of software can greatly affect how it is used 24. Programmers must prioritize making their code well-documented and readable for it to be maintained 25. Crowdsourcing identifies new problems and provides new solutions 26. Computing is rapidly and profoundly changing science and engineering 27. Computing artifacts and programs can be higher quality as a result of collaboration 28. Working in a team requires effective communication, clear responsibilities, and attention to interpersonal relationships 29. Creative ideas and technical solutions must be communicated in a clear and concise manner 30. How people present themselves affects how their work is received 31. Computer science and information technology careers offer creative job opportunities for individuals with a wide variety of backgrounds and goals 32. Parallel computing is a quickly evolving field relevant to hardware, software, and users 33. Computational thinking boosts most career paths

Focus Standards Addressed in the Unit	
3.4.10.B4	Recognize that technological development has been evolutionary, the result of a series of refinements to a basic invention.
15.4.12.B	Evaluate the impact of social, legal, ethical, and safe behaviors on digital citizenship.
15.4.12.F	Compare and contrast network environments, including the function of network devices and connectivity issues.
15.4.12.M	Evaluate the impact of emerging technologies on various career paths and provide examples of industry certifications within the field.
15.4.12.H	Use programming languages to develop logical thinking and problem solving skills.

CC.1.3.11-12.J	Acquire and use accurately general academic and domain-specific words and phrases, sufficient for reading, writing, speaking, and listening at the college and career readiness level; demonstrate independence in gathering vocabulary knowledge when considering a word or phrase important to comprehension or expression.
CC.1.4.11-12.F	Demonstrate a grade-appropriate command of the conventions of standard English grammar, usage, capitalization, punctuation, and spelling.
CC.2.1.HS.F.3	Apply quantitative reasoning to choose and Interpret units and scales in formulas, graphs and data displays.
CC.3.5.11-12.C	Follow precisely a complex multistep procedure when carrying out experiments, taking measurements, or performing technical tasks; analyze the specific results based on explanations in the text.
CC.3.5.11-12.D	Determine the meaning of symbols, key terms, and other domain-specific words and phrases as they are used in a specific scientific or technical context relevant to grades 11–12 texts and topics .
CC.3.5.11-12.G	Integrate and evaluate multiple sources of information presented in diverse formats and media (e.g., quantitative data, video, multimedia) in order to address a question or solve a problem.
CC.3.5.11-12.I	Synthesize information from a range of sources (e.g., texts, experiments, simulations) into a coherent understanding of a process, phenomenon, or concept, resolving conflicting information when possible.
CC.3.6.11-12.C	Produce clear and coherent writing in which the development, organization, and style are appropriate to task, purpose, and audience.
CC.3.6.11-12.E	Use technology, including the Internet, to produce, publish, and update individual or shared writing products in response to ongoing feedback, including new arguments or information.
CC.3.6.11-12.F	Conduct short as well as more sustained research projects to answer a question (including a self-generated question) or solve a problem; narrow or broaden the inquiry when appropriate; synthesize multiple sources on the subject, demonstrating understanding of the subject under investigation.
CC.3.6.11-12.G	Gather relevant information from multiple authoritative print and digital sources, using advanced searches effectively; assess the strengths and limitations of each source in terms of the specific task, purpose, and audience; integrate information into the text selectively to maintain the flow of ideas, avoiding plagiarism and overreliance on any one source and following a standard format for citation.
CC.3.6.11-12.H	Draw evidence from informational texts to support analysis, reflection, and research.
CC.3.6.11-12.I	Write routinely over extended time frames (time for reflection and revision) and shorter time frames (a single sitting or a day or two) for a range of discipline-specific tasks, purposes, and audiences.

Important Standards Addressed in the Unit	
CC.2.4.HS.B.6	Use the concepts of independence and conditional probability to interpret data.
15.4.12.C	Develop criteria for analyzing hardware options to meet defined needs.
CC.3.5.11-12.B	Determine the central ideas or conclusions of a text; summarize complex concepts, processes, or information presented in a text by paraphrasing them in simpler but still accurate terms.
CC.3.6.11-12.A	Write arguments focused on discipline-specific content.
CC.3.6.11-12.B	Write informative/explanatory texts, including the narration of historical events, scientific procedures/experiments, or technical processes.

Misconceptions	Proper Conceptions
1. Intelligent behavior is only useful if it exists in a robot.	1. This may be the movie screen image as envisioned by many, but intelligent behavior it is much more useful and practical when it transcends mimicking physical behavior and allows computers to make valid predictions by analyzing vast amounts of data that would be impractical for a human.

Concepts	Competencies	Vocabulary
• Moore's Law and Modeling • Intelligent Agents	• Describe the interaction between software and hardware. Use a visualization to interpret the results of a simulation. • Analyze the results of a simulation and relate the discrete nature of digital data to the results.	Agent Agent Based Modeling Calibration Complementary metal-oxide-semiconductors (CMOS) Crowd Sourcing Deterministic Discrete Component Distributed Computing Emergent Behavior Equilibrium Exponential Function Flops Law of Accelerating Returns Logic Gate Logic Table Monte Carlo Method Moore's Law Parameterization Pinout Simulation and Modeling Stochastic Transient Transistor Transistor-Transistor Logic (TTL) Validation Verification Very-large-scale Integration (VLSI) Genetic Algorithm Infinite Detail Neural Network Perceptron Self Similar Stochastic

Assessments

Engineering Journal Checks – Students will maintain a formal engineering journal to document their work throughout the course. Periodic checks will assess proper notebook format and content.

Oral Presentations – Students will periodically present their progress to their teacher and peers.

Unit Tests / Unit Projects – Each unit may include a summative written test or project. Projects may be assessed through a presentation, engineering notebook review, electronic submission, or a combination of one or more of these.

Course Portfolio – Students will compile a portfolio to document their body of work in the course

Suggested Strategies to Support Design of Coherent Instruction
Charlotte Danielson's Framework for Teaching: Domain 3 Instruction

All students will have dealt with a data set at some point, if for nothing else to calculate measures of central tendency. For many students though, this will be their first exploration of how one set of data can provide varying degrees of information and patterns. Provide multiple examples of small data sets with small numbers of variables and how those sets can be visualized in different ways. Students should also be given the opportunity to infer the structure of a data set when given only visualization(s) of that data set.

Differentiation:

- Provide graphic organizers
- Provide multiple concrete examples
- Break extended projects into smaller identifiable milestones with checkpoints along the way
- Pair stronger students with struggling students for peer assistance

Interdisciplinary Connections:

- Design process – Scientific method
- Research process – English / Social Studies
- Writing skills – English
- Graphics - Art

Additional Resources:

- ***https://csta.acm.org/Curriculum/sub/K12Standards.html*** Computer Science Teachers Association K-12 standards
- ***https://code.org/*** An open resource for additional programming references
- ***https://www.computer.org/*** Institue of Electrical and Electronics Engineers professional computer society website

Created By:
Rick Geesaman