

## Wilson Area School District Planned Course Guide

**Title of planned course:** 8th Grade Computer Science

**Subject Area:** Computer Science

**Grade Level:** 8th

**Course Description:** This course is designed to expand on 7th Grade computing knowledge including their prior knowledge of networking, physical computing, coding, troubleshooting and debugging. Students will continue to develop their programming skills using Javascript and an introduction to Python with extended opportunity to use some of their coding knowledge using small microcontroller boards such as the Circuit Playground, MicroBit or Makey Makey. Students will be encouraged to think creatively about the connection between these programming languages and real-world computer uses including how they relate to other types of machines that use the power of computing. Finally, they will again be challenged to their existing knowledge of the history and current computing trends to develop a future vision of the Fourth Industrial Revolution and The Internet of Things.

Eighth grade students will continue to practice their keyboarding skills with the goal of perfecting technique and memorizing primary keys. Touch typing is an example of cognitive automaticity, the ability to do things without conscious attention or awareness. It enables students to use higher level thinking for critical academic pursuit. Although speed and accuracy of text will be secondary, there will be a goal range of 20-30 wpm with at least 90 percent accuracy and achievement beyond their previous year's experience.

All students will be exposed to computing practices that include:

- Fostering an inclusive computing culture
- Collaborating around computing
- Recognizing and defining computational thinking
- Developing and using abstractions
- Creating computational artifacts
- Testing and refining a computational artifact
- Communicating about computing

Lessons will be presented in either Chromebook or Windows Desktop format depending on which device is most appropriate for the instruction and objectives. Students will access the assignments through Google Classroom and showcase their work using connected applications, worksheets, and finished projects.

**Time/Credit for this Course:** 36 days of instruction, practice and/or application of skills. Students in Grade 8 have scheduled computer classes for a block of 36 consecutive days during the academic year. The actual number of classes will fluctuate based on changes to the academic calendar.

**Curriculum Writing Committee:** Susan Austin

## Curriculum Map

Students will demonstrate and develop skills and knowledge of computer science and related technologies throughout this 36-day course as part of their Computer Science instruction.

### **Week 1:** (5 days)

- Keyboarding Unit
- Review of Computers & Problem Solving Unit
  - Digital Citizenship/Historical & Current Events
  - Hardware & Software--Problem Solving & Computing
  - Analyzing and Visualizing Data

### **Week 2-4:** (15 Days)

- Keyboarding Unit
- Algorithms and Programming Unit 1 (Computing Languages-Javascript)

### **Week 5-6:** (14 days)

- Keyboarding Unit
- Algorithms and Programming Unit 2 (Computing Languages-Python)

### **Week 7:** (2 days)

- Keyboarding Unit
- Future of Computing Lesson

## Planned Course Materials

**Title of planned course:** 8th Grade Computer Science

**Subject Area:** Computer Science

**Grade Level:** 8

### **Teacher and Student Resources:**

- In January 2018 the Pennsylvania State Board of Education endorsed the Computer Science Teachers Association (CTSA) K-12 standards  
<https://www.csteachers.org/Page/standards>
- Original PA State Academic Standards for Business, Computer and Information Technology [http://static.pdesas.org/content/documents/BCIT\\_standards.pdf](http://static.pdesas.org/content/documents/BCIT_standards.pdf)
- Various Computer Application and Coding Development program resources
  - Typing Programs--Typing.com
  - Common Sense Education--<https://www.common sense.org/education/>
  - [Creative Commons Media](#) --Digital Citizenship
  - Code.org Computer Science Discoveries Unit 3 - Interactive Games & Animations 23-24
  - Carnegie Mellon University CS Academy-Exploring Programming 2nd Edition (formerly CS0) Unit 1
  - Other resources like Khan Academy and CodeHS for students needing extension or remediation:
    - Khan Academy ([Computer Programming with Khan](#))
    - CodeHS (<https://codehs.com/>)
  - Sites & references for physical computing and coding microcontrollers including the Circuit Playground, Micro:bit, Scratch, & Makey Makey.
  - References from Computer Hope for student research and vocabulary.
  - EdPuzzle Computer Lessons
- Teacher created activities, worksheets, assessments
- Google Applications including Google Classroom

## Curriculum Scope & Sequence

**Title of planned course:** 8th Grade Computer Science

**Unit:** Keyboarding

**Time frame:** 5-10 minutes of each class (36 classes)

### **8th Grade Keyboarding Standards:**

- K1 Use correct posture when keyboarding and demonstrate mastery of proper keyboarding technique
- K2 Use language skills including capitalization, punctuation, spelling, word division, and the use of numbers and symbols as grade-level appropriate
- K3 Use a variety of software applications Google Slides, Docs, Sheets to produce, proofread, and correct errors within documents
- K4 Students will keyboard 30 words per minute with 95% accuracy by the end of the eighth grade course
- K5 Use one space after all punctuation marks
- K6 Keyboard from copy (documents, textbooks, or other printed sources of information)

**Essential content/objectives:** At end of the unit, students will be able to:

Students will continue building a foundation for correct typing technique by accessing all letters and basic punctuation keys, utilizing all ten fingers. (Seventh grade students will be encouraged to reach a speed of 25 wpm with 97% accuracy. Proper keyboarding technique and memorization of key locations will be valued over speed and accuracy.)

Students will be able to use correct keyboarding technique:

- Demonstrate correct use space bar, return, enter, shift, tab, Esc and Control keys.
- Demonstrate correct use of right and left hands
- Practice alphabetic keys by touch
- Demonstrate mastery of the home row keys
- Demonstrate correct fingering of basic punctuation keys
- Demonstrate proper spacing after punctuation
- Demonstrate use of the enter key, and the shift key to capitalize and access additional symbols
- Practice accessing the number keys
- Show correct body and finger positions
- Compose at the keyboard as well as proofread and correct errors within a document

**Core Activities:** Students will complete/participate in the following:

- Access specific grade level lessons in Typing.com
- Practice progressive skill based lessons in Typing.com
  - Identify the Home Row and Correct Finger Placement.
  - J, F, & Space Keys
  - U, R, & K Keys
  - D, E, & I Keys
  - C, G, & N Keys
  - Beginner Review 1
  - Personalized Practice customized to review individual problem keys
  - T, S, & L Keys
  - O, B, & A Keys

- o V, H, & M Keys
- o Period & Comma
- o Beginner Review 2
- o Personalized Practice
- o W, X, & semicolon Keys
- o Q, Y, & P Keys
- o Z and Enter Keys
- o Beginner Wrap-up
- o Personalized Practice
- o Shift Key and Capitalization
- o Basic & Intermediate Punctuation (Quotation Marks, Colons, Slashes, Question Marks)

**Instructional Methods:**

- Direct instruction
- Modeling correct keyboarding techniques
- Independent practice
- Individual student assistance (hand-over-hand)

**Materials & Resources:**

- Computer and other peripherals
- Google applications
- Typing.com Grade 8 sequence
- Keyboard Covers
- Supplemental worksheets

**Assessments:**

- Formative:
  - o Daily Assignments
  - o Observation
  - o Self-monitoring progress the Typing.com interface
- Summative
  - o Teacher reports for semester progress from Typing.com

## Curriculum Scope & Sequence

**Title of planned course:** 8th Grade Computer Science

**Unit:** Review of Computers & Problem Solving Unit

- Digital Citizenship/Historical & Current Events
- Hardware & Software--Problem Solving & Computing
- Analyzing and Visualizing Data

**Time frame:** 5 classes

**State Standards:**

- Impacts of Computing: CSTA 2-IC-20, 2-IC-21, 2-IC-22, 2-IC-23
- Computing Systems: CSTA 2-CS-03, 2-IC-23
- Data & Analysis: CSTA 2-DA-07, 2-DA-08, 2-DA-09

**Essential content/objectives:** At end of the unit, students will be able to:

- Understand and review introductory concepts from Grade 6 and 7 computer courses
- Identify the connections between human and computer interactions and brainstorm ideas for improvement
- Make decisions about computer hardware and software to accomplish specific tasks or goals given a variety of factors
- Identify the interconnected components of computing systems (hardware, software, connections) that work together for effective systems
- Consider more complex issues for troubleshooting problems
- Identify and explain how computers use, display and share data. (Since this is the first unit, within the context of initial instruction, students will also review the district AUP, including shared devices, password and security issues and digital citizenship.)

**Core Activities:** Students will complete/participate in the following:

- Demonstrate AUP knowledge and model procedures appropriate digital access & behaviors
- Review and practice good online digital citizenship
- Describe various design standards related to computing devices and applications including accessibility, ergonomics, and learnability.
- Diagnose and identify possible solutions for hardware/software problems using an understanding of the interconnections between components of functional computer systems. (Examples of troubleshooting strategies include following a troubleshooting flow diagram, making changes to software to see if hardware will work, checking connections and settings, and swapping in working components.)
- Describe the Problem Solving Process and Practice it during an unplugged group or partner activity. (Code.org CS Discoveries Unit1--Chapter 1 Activities from Lessons 1-3. For example Power Tower Activity.)
- Define a Computer and Describe the Function using activities from Code.org CS Discoveries Unit 1 --Chapter 2: Lessons 4-7.
  - What is a Computer? (Google Slide Activity)
  - Analyze & Describe Functions:
    - Input & Output

- Processing
- Storage
- Describe various ways that computers use and represent data to solve problems and create more effective ways of visualizing data. (Code.org CS Discoveries Unit 2/Chapter 1 activities from Lessons 1-7.)
- Give examples of how the same data can be represented in different ways. (Colors can be represented using binary, RGB, or HexCode. Higher level and text based coding might allow for actual words to code color---but the visual display is the same.)

**Instructional Methods:**

- Direct instruction
- Model tasks
- Discussion
- Independent practice using online applications and related activity sheets
- Scaffolding and group work
- Review

**Materials & Resources:**

- Computing Device and attached peripherals
- Google applications
- WAIS Acceptable Use Policy for the 2023-24 Academic Year
- Code.org--CS Discoveries (Problem Solving & Computing Unit 1 and Data & Society Unit 2--Updated for '23-'24)
- Common Sense Media Activities
- Creative Commons Web Resources
- EdPuzzle
- Unplugged activities

**Assessments:**

- Formative:
  - Discussions
  - Observation
  - Self Evaluations
  - Google Classroom assignments
  - Code.org Activities
- Summative
  - Completed unplugged worksheets
  - Completed online forms/guided questions

## Curriculum Scope & Sequence

**Title of planned course:** 8th Grade Computer Science

**Unit:** Algorithms and Programming Unit 1 (Computing Languages-Javascript)

**Time frame:** 15 classes

**State Standards:** Algorithms and Programming CSTA 2-AP-10, 2-AP-11, 2-AP-12, 2-AP-13, 2-AP-16, 2-AP-17, 2-AP-19

**Essential content/objectives:**At end of the unit, students will be able to:

- Create programmatic images, animations, interactive art, and games in the Interactive Animations and Games unit
- Starting off with simple, primitive shapes and building up to more sophisticated sprite-based games, students become familiar with the programming concepts and the design process computer scientists use daily
- Learn how these simpler constructs can be combined to create more complex programs
- Code an animated design using simple Javascript coding for the final project

**Core Activities:** Students will complete/participate in the following:

- Interactive Animations and Games-Chapter 1 (Lesson 1-14)
  - Programming with a Purpose-Lesson 1
    - Unplugged activity to create a prototype of an animation or game design to meet the needs of a user using the problem-solving process. (Partner or Group activity)
    - Identify features of an animation or game design that match the needs of users.
  - Plotting Shapes-Lesson 2
    - Communicate how to draw an image in Game Lab, accounting for shape position, color, and order using locations on a coordinate grid.
  - Drawing in Game Lab-Lesson 3
    - Sequence code correctly to overlay shapes.
  - Shapes and Parameters-Lesson 4
    - Use a coordinate system to place elements on the screen and utilize and explain drawing commands with multiple parameters.
  - Variables-Lesson 5
    - Identify a variable as a way to label and reference a value in a program
    - Use variables in a program to store a piece of information that is used multiple times
  - Random Numbers-Lesson 6
    - Generate and use random numbers in a program.
    - Update a value stored in a variable.
  - Sprites-Lesson 8
    - Create and use a sprite (complex sets of images).
  - Sprites Properties-Lesson 9



- Access and manipulate sprite properties to program more interesting behaviors incorporating dot notation to update sprites properties.
- Text Elements-Lesson 10
  - Place text on the screen using a coordinate plane and use various arguments to control how text is displayed on a screen.
- The Draw Loop-Lesson 12
  - Use the draw loop in combination with the randomNumber() command, shapes, and sprites to make simple animations.
- Sprite Movement-Lesson 13
  - Use the counter pattern to increment or decrement sprite properties to alter sprites and their behavior to achieve specific movements.
- Design an Animated Program-Lesson 14
  - Apply learned programming concepts to create an individualized program using a structured planning process.
  - Use self-evaluation and project presentation or description with completed project/program. (For example: Outline key features of program development and explain their choices made using a storyboard, journaling, or summative presentation.)
  - Peer Evaluation for student projects

**Instructional Methods:**

- Direct instruction
- Model tasks
- Discussion
- Independent practice using online applications and related activity sheets
- Scaffolding and group work
- Review

**Materials & Resources:**

- Computing Device and attached peripherals
- Google applications
- Code.org--CS Discoveries Unit 3 (Interactive Animations and Games)
- Unplugged activities and worksheets to support lessons

**Assessments:**

- Formative:
  - Discussions
  - Observation
  - Self Evaluations
  - Peer Evaluations
  - Google Classroom assignments
  - Code.org activities
- Summative
  - Completed unplugged worksheets
  - Completed lesson sections within Code.org Unit 3
  - Projects

## Curriculum Scope & Sequence

**Title of planned course:** 8th Grade Computer Science

**Unit:** Algorithms and Programming Unit 2 (Computing Languages-Python)

**Time frame:** 14 classes

**State Standards:** Algorithms and Programming CSTA 1A-AP-14, 1A-AP-10, 1B-AP-10, 1B-AP-09, 2-AP-13, 2-AP-17

**Essential content/objectives:** At end of the unit, students will be able to:

- Gain experience with the Python computing language
- Bridge their intermediate school computer programming experience and course offerings at the high school level
- Students will have the opportunity to continue their Computer Science experience with the Carnegie Mellon Academy curriculum using the Python language for High School

**Core Activities:** Students will complete/participate in the following:

- Effectively access the Interactive CMU CS Academy application.
- Recall and discuss the Coordinate Grid --compare and contrast Code.org to CMU CS
- Drawing with Shapes with Python-Unit 1
  - Introduction-Lesson 1
    - Draw a circle
    - Debug with the inspection tool
    - Draw a landscape
    - Practice
  - Stars & Gradients-Lesson 2
    - Create a star
    - Use color shading or gradients
    - Combine techniques to give dimension/depth
    - Practice
  - Rectangles & Opacity-Lesson 3
    - Create a rectangle
    - Change size using width & height
    - Calculate the width & height
    - Add fill color and manipulate opacity
    - Practice
  - More Shapes, Properties, Text-Lesson 4
    - Ovals
    - Lines
    - Labels (text and properties)
    - Practice
  - Wrap-up-Lesson 5
    - Final Practice Sets
    - Challenge Exercises
  - Creative Expression (Individual or Group)-Lesson 6

- Final Assessment-Lesson 7
- Use self-evaluation and project presentation or description with completed project/program. (For example: Outline key features of program development and explain their choices made using a storyboard, journaling, or summative presentation.)
- Peer Evaluation for student Projects
- If students finish each lesson in one class period, then Unit will include Basic Animations. (See extension plan.)

**Extension Plan:**

- Basic Animations with Python-Unit 2
  - Using the Mouse-Lesson 6
    - “onMousePress”
    - Practice
  - Animating With the Mouse-Lesson 7
    - Naming Shapes
    - “onMouseMove”
    - Practice
  - Other types of Animations-lesson 8
    - “onStep”
    - Special Motion
    - Practice
  - Wrap-up-Lesson 9
    - Functions & Variables (for advanced students)
    - Playing Catch
    - Challenges
  - Wrap-up-Lesson 5
    - Final Practice Sets
    - Challenge Exercises
  - Creative Expression (Individual or Group)
  - Final Assessment

**Instructional Methods:**

- Direct instruction
- Model tasks
- Discussion
- Independent practice using online applications and related activity sheets
- Scaffolding and group work
- Review

**Materials & Resources:**

- Computing Device and attached peripherals
- Google Applications
- CMS CS Academy-Exploring Programming with Python (formerly CS0)
- Common Sense Media Activities
- Unplugged activities or worksheets

**Assessments:**

- Formative:
  - Discussions
  - Observation
  - Self Evaluations
  - Peer Evaluations
  - Google Classroom assignments
  - Daily activities in CMU CS application
- Summative
  - Completed unplugged worksheets
  - Completed online forms/guided questions
  - Projects

## Curriculum Scope & Sequence

**Title of planned course:** 8th Grade Computer Science

**Unit:** The Future of Computing

**Time frame:** 2 classes

**State Standards:** Impacts of Computing: CSTA 2-IC-20, 2-IC-21

**Essential content/objectives:** At end of the unit, students will be able to:

- Analyze how computing technology has impacted our world including lifestyles, resources, and careers
- Compare and contrast how various computing technologies are both positive and negative
- Consider recent local, state, national, and global events and correlate those with digital technologies including privacy, communication, and automation

**Core Activities:** Students will complete/participate in the following:

- Analyze and discuss how computing technologies have impacted the world and influenced cultural and societal practices. (the Internet, AI, GPS, etc.)
- Define and model at least one of these concepts with respect to computing technologies:
  - Crowdsourcing
  - Artificial Intelligence
  - Social Engineering
  - Phishing & Spoofing & Hacking
  - Facial Recognition
  - Accessibility
  - Sustainability
- Using a Goggle application of choice, showcase through collaboration and an understanding of a recent/current event impact and make a clear connection to the future of computing and/or technology impacts. The work should focus on a reasonable prediction of the future and must be presented with respect to intellectual property permission and appropriate attribution.
- Collaborate and seek feedback from others with the purpose to make improvements to projects.

**Instructional Methods:**

- Direct instruction
- Model tasks
- Discussion
- Independent practice using online applications and related activity sheets
- Scaffolding and group work
- Review

**Materials & Resources:**

- Computing Device and attached peripherals

- Google applications
- WAIS Acceptable Use Policy for the 2023-24 Academic Year
- Common Sense Media Activities
- EdPuzzle
- Unplugged activities

**Assessments:**

- Formative:
  - Discussions
  - Observation
  - Self Evaluations
  - Peer Evaluations
  - Google Classroom assignments
- Summative
  - Completed unplugged worksheets
  - Completed online forms/guided questions

## **PA Computer Science Standards**

### **Computing Systems**

- Devices: 2-CS-01 Recommend improvements to the design of computing devices, based on an analysis of how users interact with the devices
- Hardware & Software: 2-CS-02 Design projects that combine hardware and software components to collect and exchange data
- Troubleshooting: 2-CS-03 Systematically identify and fix problems with computing devices and their components

### **Networks and the Internet**

- Network Communication & Organization: 2-NI-04 Model the role of protocols in transmitting data across networks and the Internet
- Cybersecurity: 2-NI-05 Explain how physical and digital security measures protect electronic information

### **Data and Analysis**

- Storage: 2-DA-07 Represent data using multiple encoding schemes
- Collection Visualization & Transformation: 2-DA-08 Collect data using computational tools and transform the data to make it more useful and reliable
- Inference & Models: 2-DA-09 Refine computational models based on the data they have generated

### **Algorithms and Programming**

- Algorithms: 2-AP-10 Use flowcharts and/or pseudocode to address complex problems as algorithms
- Variables: 2-AP-11 Create clearly named variables that represent different data types and perform operations on their values
- Control: 2-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals
- Modularity:
  - 2-AP-13 Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs
  - 2-AP-14 Create procedures with parameters to organize code and make it easier to reuse
- Program Development:
  - 2-AP-15 Seek and incorporate feedback from team members and users to refine a solution that meets user needs
  - 2-AP-16 Incorporate existing code, media, and libraries into original programs, and give attribution
  - 2-AP-17 Systematically test and refine programs using a range of test cases
  - 2-AP-18 Distribute tasks and maintain a project timeline when collaboratively developing computational artifacts

- 2-AP-19 Document programs in order to make them easier to follow, test, and debug

### **Impacts of Computing**

- Culture:
- Social Interactions:
  - 2-IC-20 Compare tradeoffs associated with computing technologies that affect people's everyday activities and career options
  - 2-IC-21 Discuss issues of bias and accessibility in the design of existing technologies
  - 2-IC-22 Collaborate with many contributors through strategies such as crowdsourcing or surveys when creating a computational artifact
- Safety Law & Ethics: 2-IC-23 Describe tradeoffs between allowing information to be public and keeping information private and secure

### **Other Related PA State Standards/Anchors**

- 15.3.8.M. Demonstrate proper etiquette when networking either face-to-face or online
- 15.3.8.S. Apply appropriate electronic communication based on message requirements
- 15.3.8.T. Discuss the rules of digital citizenship
- 15.3.12.W. Use electronic communication with peers and/or educators to produce a work product
- 15.4.8.B. Interpret and apply appropriate social, legal, ethical, and safe behaviors of digital citizenship
- 15.4.8.C. Compare and contrast peripheral devices of computing systems for specific needs
- 15.4.8.D. Create projects using emerging input technologies
- 15.4.8.G. Create an advanced digital project using appropriate software/application for an authentic task
- 15.4.8.H. Explain the differences between a scripting language and a coding language
- 15.4.8.I. Solve a problem with an algorithm
- 15.4.8.J. Explain the basic differences between encoding and decoding
- 15.4.8.K. Create a multimedia project using student created digital media
- 15.6.8.L. Evaluate the accuracy and bias of online sources of information; appropriately cite online resources
- 15.4.8.D. Create projects using emerging input technologies
- 15.4.8.B. Interpret and apply appropriate social, legal, ethical, and safe behaviors of digital citizenship
- 15.4.8.A. Analyze the influence of emerging technologies on daily life

### **8th Grade Keyboarding Standards:**

- K1 Use correct posture when keyboarding and demonstrate proper keyboarding technique for each letter of the alphabet
- K2 Keyboard word lists and sentences using correct posture and proper keyboarding technique
- K3 Use language skills including capitalization, punctuation, spelling, word division, and the use of numbers and symbols as grade-level appropriate
- K6 Use a variety of software applications to produce, proofread, and correct errors within documents
- K5 Students will keyboard 27 words per minute with 95% accuracy by the end of the fifth grade



- K6 Use one space after all punctuation marks
- K7 Keyboard from copy (documents, textbooks, or other printed sources of information)