Cover Page

Content Area: **Sample Content Area**
Course(s): **Computer Science (IS)**
Time Period: **1st Marking Period**
Length: **45 days**
Status: **Not Published**

Title Page, Table of Contents, Statement of Purpose

Sayreville Public Schools Curriculum

Computer Science

8th Grade

Table of Contents

Statement of Purpose

Summary of the Course: Computer Science Grade 8 is a course designed for general education students and special education students to work collaboratively completing projects and assignments utilizing Code.org. Computer Science Discoveries (CS Discoveries) is an introductory computer science course that empowers students to create authentic artifacts and engage with computer science as a medium for creativity, communication, problem-solving, and fun.
In order to demonstrate a cohesive and complete implementation plan the following general suggestions are provided:
● Homework, when applicable
● Differentiated Instruction
● Varied assessments
● Modifications to the curriculum should be included that address students with Individualized Educational Plans (IEP), English Language Learners (ELL), those requiring other modifications (504 plans), as well as extension exercises for gifted learners.

Unit 1: Design Process

Content Area: **Business**
Course(s):
Time Period: **1st Marking Period**
Length: **10 Weeks**
Status: **Not Published**

Summary of the Unit

This unit introduces the broader social impacts of computing. Through a series of design challenges, you will learn how to better understand the needs of others while developing a solution to a problem. This unit also consists of an iterative team project, during which teams have the opportunity to identify a need that they care about, prototype solutions both on paper and in App Lab, and test solutions with real users to get feedback and drive further iteration.

Enduring Understanding

By the end of the unit, students should see the design process as a form of problem solving that prioritizes the needs of a user. They should be able to identify user needs and assess how well different designs address them. Students should leave the unit with a basic understanding of other roles in software development and how to use what they have learned about computer science as a tool for social impact.

Essential Questions

- How do computer scientists identify the needs of their users?
- How can we ensure that a user's needs are met by our designs?
- What processes will best allow us to efficiently create, test, and iterate upon our design?
- How do teams effectively work together to develop software?
- What roles beyond programming are necessary to design and develop software?
- How do designers incorporate feedback into multiple iterations of a product?

Summative Assessment and/or Summative Criteria

- Students will complete assessment covering topics from Unit 1

Resources

- CS Discoveries Curriculum (www.code.org)

- AppLab Online IDE
- Throughout the course, students will also use a number of online articles and videos from sources such as The New York Times (www.nytimes.com), Wikipedia (www.wikipedia.org), CS Bits and Bytes (http://www.nsf.gov/cise/csbytes/), Logic.ly (www.logic.ly), YouTube (www.youtube.com), and CS Unplugged (http://csunplugged.org).

Unit Plan

Topic/Selection Timeframe	General Objectives	Instructional Activities	Benchmarks/A
Lesson 1: Designing with Empathy (1 day)	SWBAT • Explore the relationship between users, their needs, and the design of objects they use.	• The class explores a variety of different shoe designs to consider design choices.	• Post Lesson I
Lesson 2: Understanding Your User (1 day)	SWBAT • Explore how different users might react to a variety of products.	• Using user profiles, students will be role playing as a different person, each member of the class will get to experience designs through someone else's eyes.	• Post Lesson I
Lesson 3: User-Centered Design - Define and Prepare	SWBAT • Use the design process to come up with ideas for smart clothing.	• Today's lesson focuses on brainstorming users and ideas that will meet their needs. Over the course of both lessons, students will brainstorm ideas, identify users, and finally propose a design. This activity serves as the first of several opportunities for students to practice designing a solution for the needs of others.	• Post Lesson I

Lesson 4: User-Centered Design - Try and Reflect (1 day)	SWBAT • Use the design process to come up with ideas for smart clothing	• Today's lesson focuses on creating a design and reflecting on how well it meets the needs of users. • Students will brainstorm ideas, identify users, and finally propose a design. • This activity serves as the first of several opportunities for students to practice designing a solution for the needs of others.	• Post Lesson I
Lesson 5: User Interfaces (1 day)	SWBAT • Determine how a paper prototype can be used to test and get feedback before writing any code	• The class tests and provides feedback on an app prototype made of paper.	• Post Lesson I
Lesson 7: Identifying User Needs (1 day)	SWBAT • Brainstorm ideas for an app that can solve a problem in our community	• Using interviews from different users, students identify needs and interests that they can use to design an app for these people in their community.	• Post Lesson I
Lesson 8: Project - Paper Prototype (3 days)	SWBAT • Brainstorm app ideas to address the needs of their users.	• Using the interview information from the previous lesson, students come up with app ideas to address the needs of their users. To express those ideas, and test out their effectiveness, students create and test paper prototypes.	• Post Lesson I
Lesson 9: Designing Apps for Good	SWBAT • Explore the social impact of creating applications	• To kick off the app design project, the class organizes into teams and starts exploring app topics. Several examples of socially impactful apps serve as inspiration for the project.	• Post Lesson I

Lesson 11: Exploring UI Elements	SWBAT • Explore methods used by developers to test programs	• Paper prototypes allow developers to quickly test ideas before investing a lot of time writing code. In this lesson, teams explore some example apps created in App Lab and use these examples to help inform the first paper prototypes of their apps.	• Post Lesson I
Lesson 12: Build a Paper Prototype	SWBAT • Use design techniques to decide how the user will navigate a project	• In teams, students will create a paper prototype for the app they've been developing. Each team member will create a different screen and design how the user will navigate between each screen.	• Post Lesson I
Lesson 14: Design Mode in App Lab	SWBAT • Use CS Industry techniques to design a prototype	• Teams now move to App Lab to build the next iteration of their apps. This lesson focuses on how to use Design Mode in App Lab to create digital prototypes for their apps.	• Post Lesson I
Lesson 15: Build a Digital Prototype	SWBAT • Divide and discuss project roles to collaborate in a group	• Using the drag-and-drop Design Mode, each team member builds out at least one page of their team's app, responding to the feedback received in the previous round of testing.	• Post Lesson I
Lesson 16: Events in App Lab	SWBAT • Discover App Lab components to import new screens into our apps	• Building on the previous lesson, we learn how to import new screens into our apps and link them together using buttons and events to complete the Recycle Finder app we started in an earlier	• Post Lesson I

		lesson.	
Lesson 18: Testing the App	SWBAT • Explore methods used by developers to test programs	• In this lesson, teams run another round of user testing with their interactive prototype. Feedback gathered from this round of testing will inform the final iteration of the digital prototype.	• Post Lesson I
Lesson 21: Project - App Presentation (3 day)	SWBAT • Articulate the problem they are solving to a group	• Each team prepares a presentation to "pitch" the app they've developed. This is the time they can share the struggles, triumphs, and plans for the future.	• Post Lesson I

CS.6-8.DA	Data & Analysis
CS.6-8.NT	Nature of Technology
CS.6-8.EC	Ethics & Culture
CS.6-8.CS	Computing Systems
CS.6-8.ED	Engineering Design
CS.6-8.ITH	Interaction of Technology and Humans
CS.6-8.AP	Algorithms & Programming
CS.6-8.IC	Impacts of Computing

Suggested Modifications for Special Education, ELL and Gifted Students

*Consistent with individual plans, when appropriate.

Suggested Technological Innovations/Use

- Students should use online resources to research topics covered
- Students will use integrated IDE to test and execute code

- Students will use Scratch online documentation resource

Cross Curricular/21st Century Connections

English Language Arts • Journal writing • Close reading of industry-related content • Keep a running word wall of industry vocabulary

Social Studies • Research the history of a given industry/profession • Research prominent historical individuals in a given industry/profession • Use historical references to solve problems

World Language • Translate industry-content • Create a translated index of industry vocabulary • Generate a translated list of words and phrases related to information technology

Math • Compare probability and the use of random numbers in computer programming. • Track various career opportunities or among of individuals currently occupying careers

Fine & Performing Arts • Create a poster recruiting young people to focus their studies on a career in Information Technology

Science • Research the environmental impact of a given career or industry • Research latest developments in Information technology • Investigate applicable-careers in STEM fields

Unit 2 - Data and Society

Content Area: **Business**
Course(s):
Time Period: **1st Marking Period**
Length: **10 Weeks**
Status: **Not Published**

Summary of the Unit

This unit is about the importance of data in solving problems and highlights how computers can help in this process. This unit explores different systems used to represent information in a computer and the challenges and tradeoffs posed by using them. This unit also teaches how collections of data are used to solve problems, and how computers help to automate the steps of this process. It concludes by considering how the data problem solving process can be applied to an area of your choosing.

Enduring Understandings

Students should have a broad understanding of the role of data and data representation in solving information problems. Students should also be able to design and implement a data-based solution to a given problem and determine how the different aspects of this problem solving process could be automated.

Essential Questions

- Why is representation important in problem solving?
- What features does a representation system need to be useful?
- How does data help us to solve problems?
- How do computers and humans use data differently?
- What kinds of real world problems do computers solve by using data?

Summative Assessment and/or Summative Criteria

- Students will complete an assessment covering topics from Unit 2.

Resources

- CS Discoveries Curriculum (www.code.org)
- AppLab Online IDE
- Throughout the course, students will also use a number of online articles and videos from sources such as The New York Times (www.nytimes.com), Wikipedia (www.wikipedia.org), CS Bits and Bytes (http://www.nsf.gov/cise/csbytes/), Logic.ly (www.logic.ly), YouTube (www.youtube.com), and CS Unplugged (http://csunplugged.org).

Unit Plan

Topic/Selection Timeframe	General Objectives	Instructional Activities	Benchmarks/A
Representation Matters (1 Day)	SWBAT • compare responses and discuss how the different representations	• Groups use a data set to make a series of meal recommendations for people with various criteria. • Afterward, groups compare their responses and discuss how the different representations of the meal data affected how they were able to solve the different problems.	• Post Les

Patterns and Representation (1 Day)	SWBAT • create systems and messages • class discussion	• Groups create systems that can represent any letter in the alphabet using only a single stack of cards. • They then create messages with their systems and exchange with other groups to ensure the system worked as intended. • Finally, the class discusses commonalities between working systems while recognizing that there are many possible working solutions.	• Post Less
Keeping Data Secret (1 Day)	SWBAT • explore how data is represented	• Students continue to explore how data is represented in a punchcard, and begin considering whether some data should be protected from public view because it is too personal or sensitive. • Then, they learn a binary encryption system that lets them encrypt and decrypt data in their punchcards.	• Post Less

Create a Representation (1 Day)	SWBAT • designs structures	• The class designs structures to represent their perfect day using the binary representation systems they've learned in this chapter. • students decide how a punch card of bytes of information will be interpreted to represent those pieces of information. • they use the ASCII, binary number, and image formats they have learned to represent their perfect days and try to decipher what a partner's perfect day is like.	• Post Less
Problem Solving and Data (1 Day)	SWBAT • find and use data	• The class is tasked with deciding what a city most needs to spend resources on. They must find and use data from the internet to support their decision.	• Post Less
Structuring Data (1 Day)	SWBAT • present data in different ways	• The class first looks at how presenting data in different ways can help people to understand it better. • students look at which parts of this process can be automated, and which parts need a human.	• Post Less
Interpreting Data (1 Day)	SWBAT • present data in different ways	• Students look at a cake preference survey and discuss how knowing the relationship between cake and icing preference helps them better decide which combination to recommend. • Students are then introduced to cross tabulation, which allows them to graph relationships to different preferences. They use this technique to find relationships in a preference survey, then brainstorm the different types of problems that this process could help solve.	• Post Less

Making Decisions with Data (1 Day)	SWBAT • practice the data problem solving process	• Students will have a discussion about how different people could draw different conclusions from the same data, and how collecting different data might have affected the decisions they made.	• Post Less
Automating Data Decisions (1 Day)	SWBAT • create an algorithm that suggests a vacation spot • create rules that a computer could use to make this decision automatically	• Students share their rules and test their rules with the class data. • Next, they use data from their classmates to test whether their rules would make the same decision that a person would. • Then they will have a discussion about the benefits and drawbacks of using computers to automate the data problem solving process.	• Post Less
Problem Solving with Big Data (1 Day)	SWBAT • collect and use data to solve problems in the real world	• Students look at three scenarios that could be solved using data and brainstorm the types of data they would want to use to solve each problem, as well as strategies they could use to collect the data. Each scenario also includes a video about a real-world service that has solved a similar problem with data.	• Post Less
Project - Make a Recommendation (1 Day)	SWBAT • design ways to use data to make recommendations or predictions to help solve a problem • brainstorm problems, perform simple research, and define a problem of their choosing	• Students brainstorm problems, perform simple research, and define a problem of their choosing. • They then decide what kind of data they want to collect, how it could be collected, and how it could be used, before exchanging feedback and giving a final presentation.	

CS.6-8.DA	Data & Analysis
CS.6-8.NT	Nature of Technology
CS.6-8.EC	Ethics & Culture
CS.6-8.CS	Computing Systems

CS.6-8.ETW	Effects of Technology on the Natural World
CS.6-8.ED	Engineering Design
CS.6-8.ITH	Interaction of Technology and Humans
CS.6-8.AP	Algorithms & Programming
CS.6-8.IC	Impacts of Computing

Suggested Modifications

*Consistent with individual plans, when appropriate.

Suggested Technological Innovations/Use

- Students should use online resources to research topics covered
- Students will use integrated IDE to test and execute code
- Students will use Scratch online documentation resource

Cross-Curricular/21st Century Connections

English Language Arts • Journal writing • Close reading of industry-related content • Keep a running word wall of industry vocabulary

Social Studies • Research the history of a given industry/profession • Research prominent historical individuals in a given industry/profession • Use historical references to solve problems

World Language • Translate industry-content • Create a translated index of industry vocabulary • Generate a translated list of words and phrases related to information technology

Math • Compare probability and the use of random numbers in computer programming. • Track various career opportunities or among of individuals currently occupying careers

Fine & Performing Arts • Create a poster recruiting young people to focus their studies on a career in Information Technology

Science • Research the environmental impact of a given career or industry • Research latest developments in Information technology • Investigate applicable-careers in STEM fields

Unit 3: Physical Computing

Content Area: **Business**
Course(s):
Time Period: **1st Marking Period**
Length: **10 Weeks**
Status: **Not Published**

Summary of the Unit

This unit explores the role of hardware platforms in computing and how different sensors can provide more effective input and output than the traditional keyboard, mouse, and monitor. Using App Lab, you'll develop programs that utilize the same hardware inputs and outputs that you see in the smart devices, looking at how a simple rough prototype can lead to a finished product.

Enduring Understanding

By the end of the unit, students should be able to design and build a physical computing device that integrates hardware inputs and outputs with software. Students should leave the unit feeling equipped to use physical computing to solve problems in fun and innovative ways.

Essential Questions

- How does software interact with hardware?
- How can computers sense and respond to their environment?
- What kind of information can be communicated with hardware outputs?
- How can complex real-world information be represented in code?
- How can simple hardware be used to develop innovative new products?

Summative Assessment and/or Summative Criteria

- Students will complete assessment covering topics from Unit 3

Resources

- CS Discoveries Curriculum (www.code.org)
- AppLab Online IDE
- Throughout the course, students will also use a number of online articles and videos from sources such as The New York Times (www.nytimes.com), Wikipedia (www.wikipedia.org), CS Bits and Bytes (http://www.nsf.gov/cise/csbytes/), Logic.ly (www.logic.ly), YouTube (www.youtube.com), and CS Unplugged (http://csunplugged.org).

Unit Plan

Topic/Selection Timeframe	General Objectives	Instructional Activities	Benchmarks/A
Lesson 1: Innovations in Computing (1 day)	SWBAT • research into interesting innovations in computing	• Students explore a wide variety of new and innovative computing platforms while expanding their understanding of what a computer can be. • Students will look back on the devices they encountered in this lesson as they develop their own physical computing devices.	• Post Lesson I
Lesson 2: Designing Screens with Code (1 day)	SWBAT • think in new ways about how users interact with apps • build up simple apps that only require a single screen	• reading and changing the content on the screen of an app, the class starts to build apps that only need a single screen. Even with just one screen, students can begin to see that these techniques allow for lots of user interaction and functionality.	• Post Lesson I

Lesson 4: Input Unplugged (1 Day)	SWBAT • experience two different ways that an app can collect input from a user, while learning more about the event-driven programming model used in App Lab	• With App Lab, students will explore how a handful of different programs written in both Game Lab and App Lab handle taking input from the user	• Post Lesson I
Lesson 6: Getting Properties (1 day)	SWBAT • use getProperty and setProperty together with the counter pattern to make elements move across the screen • use a new screen element, the slider, and a new event trigger, onChange	• Students will practice using the getProperty and setProperty blocks to determine what the user has input in various user interface elements.	• Post Lesson I
Lesson 10: Arrays and Color LEDs (1 day)	SWBAT • learn how to access and control each LED in an array individually, preparing them to access multiple LEDs t	• Students will control the color and intensity of each LED, then use what they have learned to program light patterns to create a light show	• Post Lesson I
Lesson 14: Functions with Parameters (1 day)	SWBAT • create and call functions with parameters to control multiple elements on a screen	• Students look at examples of parameters within user-created functions in App Lab	• Post Lesson I

CS.6-8.DA	Data & Analysis
CS.6-8.NT	Nature of Technology

CS.6-8.EC	Ethics & Culture
CS.6-8.CS	Computing Systems
CS.6-8.ED	Engineering Design
CS.6-8.ITH	Interaction of Technology and Humans
CS.6-8.AP	Algorithms & Programming
CS.6-8.IC	Impacts of Computing

Suggested Modifications for Special Education, ELL and Gifted Students

*Consistent with individual plans, when appropriate.

Suggested Technological Innovations/Use

- Students should use online resources to research topics covered
- Students will use integrated IDE to test and execute code
- Students will use Scratch online documentation resource

Cross Curricular/21st Century Connections

English Language Arts • Journal writing • Close reading of industry-related content • Keep a running word wall of industry vocabulary

Social Studies • Research the history of a given industry/profession • Research prominent historical individuals in a given industry/profession • Use historical references to solve problems

World Language • Translate industry-content • Create a translated index of industry vocabulary • Generate a translated list of words and phrases related to information technology

Math • Compare probability and the use of random numbers in computer programming. • Track various career opportunities or among of individuals currently occupying careers

Fine & Performing Arts • Create a poster recruiting young people to focus their studies on a career in Information Technology

Science • Research the environmental impact of a given career or industry • Research latest developments in Information technology • Investigate applicable-careers in STEM fields