codelicious™
Code builds confidence

CSTA Educational Standards Mapping

Overview of Curriculum
Grades 3-5 and 6-8
V 2.0

1. Introduction and Purpose

Codelicious is on a mission to help students discover how to think about and speak the language of technology. With hundreds of thousands of coding jobs going unfilled each year, Codelicious aims to dramatically increase the funnel of youth coders for tomorrow's tech workforce, by providing coding curriculum for youth.

Codelicious recognizes the challenges of teaching code in schools…

Ever-evolving syllabus
Best practices in code change and evolve at the speed of technology. Traditional books and lessons become outdated quickly. The need to keep coding and computer science curriculum updated at regular intervals takes away from the time educators can spend on the learning needs of their students.

Student Engagement
Teaching problem solving and computer science to high school students is more than learning coding skills by sitting in front of a computer. Problem-solving and computer science are best learned by engaging students through their different learning styles and reinforcing concepts through off-line projects and learning opportunities.

…and provides a solution designed to support educators and engage elementary and middle school students.

Codelicious provides instant access to cloud-based, collaborative and interactive curriculum. And, because it resides in the cloud, our curriculum can be kept current with the speed of technology. Codelicious offers elementary and middle school educators and administrators not only a savings of time and money, but a turnkey solution uniquely built to transform the problem solving and coding experience they can give their students.

2. Curriculum Overview

Codelicious currently offers full courses in programming, graphics, and engineering. Throughout these courses, elementary and middle school students will master the appropriate CSTA standards. Codelicious is currently writing and quality checking curriculum for grades k-2, with a launch date of Fall 2020. Our current courses include:

- Introduction to Computational Thinking with Scratch
- Introduction to Web Development with Javascript
- Advanced Web Development with Javascript
- Introduction to Computational Thinking with Java
- Introduction to Computer Science
- Introduction to Video Game Development
- Introduction to Graphic Design with Photoshop
- Introduction to App Design
- Introduction to Python

3. Standards Map – Codelicious Courses Overview for Grades 3 – 5

The following is a complete list of Computer Science Teacher Association (CSTA) standards for grades 3 through 5 mapped to our curriculum.

Computing Systems	
Educational Standard	**How Standard is Met**
1B-CS-01 Describe how internal and external parts of computing devices function to form a system.	Students will be introduced to different types of hardware and software devices. Students will gain hands-on experience building various types of circuits and devices.
1B-CS-02 Model how computer hardware and software work together as a system to accomplish tasks.	Students will understand the concepts of how computers require specific information to accomplish tasks through unplugged activities and discussions.
1B-CS-03 Determine potential solutions to solve simple hardware and software problems using common troubleshooting strategies.	Through hardware and coding activities, students will apply and discuss troubleshooting strategies.

Networks and the Internet	
Educational Standard	**How Standard is Met**
1B-NI-04 Model how information is broken down into smaller pieces, transmitted as packets through multiple devices over networks and the Internet, and reassembled at the destination.	Through lessons about the STEM career Network Administrator, students will learn how the internet functions.
1B-NI-05 Discuss real-world cybersecurity problems and how personal information can be protected.	Students will brainstorm how personal information can be protected through cybersecurity, through real-world discussions.

Data Analysis	
Educational Standard	**How Standard is Met**
1B-DA-06 Organize and present collected data visually to highlight relationships and support a claim.	Students will practice planning, organizing, and presenting data through graphic organizers and flow charts.
1B-DA-07 Use data to highlight or propose cause-and-effect relationships, predict outcomes, or communicate an idea.	Students will develop algorithms to trace and predict the outcome of their code while frequently testing and debugging it.

codelicious™
Code builds confidence

CSTA Educational Standards Mapping

Overview of Curriculum
Grades 3-5 and 6-8
V 2.0

Algorithms and Programming	
Educational Standard	**How Standard is Met**
1B-AP-08 Compare and refine multiple algorithms for the same task and determine which is the most appropriate.	Through unplugged and coding activities, students will write their own algorithms that involve loops, functions, and conditionals.
1B-AP-09 Create programs that use variables to store and modify data.	Students will develop coding projects that utilize various computer science skills including storing and calling variables.
1B-AP-10 Create programs that include sequences, events, loops, and conditionals.	Students will develop programs that utilize sequences, events, loops, and conditionals throughout most lessons, whether it is through coding or unplugged activities.
1B-AP-11 Decompose (break down) problems into smaller, manageable subproblems to facilitate the program development process.	Through unplugged and coding activities, students will decompose problems into more manageable steps and subproblems.
1B-AP-12 Modify, remix, or incorporate portions of an existing program into one's own work, to develop something new or add more advanced features.	Coding activities utilize different platforms, such as Scratch and Godot, to modify, remix, and incorporate parts of existing programs to make them their own.
1B-AP-13 Use an iterative process to plan the development of a program by including others' perspectives and considering user preferences.	Students will work on coding projects using an iterative method and collaborate to enhance their projects.
1B-AP-14 Observe intellectual property rights and give appropriate attribution when creating or remixing programs.	In several activities, students discuss the meaning of copyright and intellectual rights. They additionally give credit to the interface used (Scratch, Godot, etc.) in all coding projects.
1B-AP-15 Test and debug (identify and fix errors) a program or algorithm to ensure it runs as intended.	Throughout coding activities, students will collaborate to debug their algorithms and projects to ensure they run as intended.
1B-AP-16 Take on varying roles, with teacher guidance, when collaborating with peers during the design, implementation, and review stages of program development.	Students will collaborate with peers on programming and debugging while developing their coding projects.
1B-AP-17 Describe choices made during program development using code comments, presentations, and demonstrations.	All coding projects encourage students to comment, discuss, and present their work for classmates and peers.

Impacts of Computing	
Educational Standard	**How Standard is Met**
1B-IC-18 Discuss computing technologies that have changed the world, and express how those technologies influence, and are influenced by, cultural practices.	Students will participate in discussions of various STEM careers as they explore how computing technologies have influenced the world.
1B-IC-19 Brainstorm ways to improve the accessibility and usability of technology products for the diverse needs and wants of users.	Through discussions of digital ethics, students will brainstorm ways to improve their coding activities for future users.
1B-IC-20 Seek diverse perspectives for the purpose of improving computational artifacts.	Through discussions and debugging, students will seek opinions from classmates and teachers in the hopes of improving their scratch programs.
1B-IC-21 Use public domain or creative commons media, and refrain from copying or using material created by others without permission.	Students utilize an open-source, block-based programming language called Scratch to develop their own games.

4. Standards Map – Codelicious Courses Overview for Grades 6 – 8

The following is a complete list of Computer Science Teacher Association (CSTA) standards for grades 6 through 8 mapped to our curriculum.

Computing Systems:	
Educational Standard	**How Course Meets Standard**
2-CS-01 Recommend improvements to the design of computing devices, based on an analysis of how users interact with the devices.	With the opportunity to design their own program from start to finish, students will have the chance to explore any areas in which they can improve devices or computing systems. As they code, they will consider the user's experience.
2-CS-02 Design projects that combine hardware and software components to collect and exchange data.	Students will examine how hardware elements interact with software. They will specifically observe and identify how computers require specific instructions in order to perform the desired task.
2-CS-03 Systematically identify and fix problems with computing devices and their components.	Through coding activities, students will troubleshoot their program. They will be working on computers to write their code, presenting opportunities to problem-solve any problems with the computer itself.

Networks and the Internet:	
Educational Standard	**How Course Meets Standard**
2-NI-04 Model the role of protocols in transmitting data across networks and the Internet.	By exploring the career of Network Administrator, students will gain an understanding of how the internet functions.
2-NI-05 Explain how physical and digital security measures protect electronic information.	Students will discuss ways in which they can protect their personal information, focusing on strategies to strengthen security.
2-NI-06 Apply multiple methods of encryption to model the secure transmission of information.	Through discussions about privacy, students will brainstorm ways to keep their data safe. They will also practice using ciphers, exploring unique ways to protect their information.

Data and Analysis:	
Educational Standard	**How Class Meets Standard**
2-DA-07 Represent data using multiple encoding schemes.	Students will explore ways to represent data, specifically how computers translate data into binary. They will then practice representing different data in binary.
2-DA-08 Collect data using computational tools and transform the data to make it more useful and reliable.	Students will use spreadsheets to read and manipulate data. Discussions throughout the activity will allow students to analyze why spreadsheets are useful when working with data.
2-DA-09 Refine computational models based on the data they have generated.	When tasked with designing their own project from start to finish, students will be challenged to revise, edit, and refine their code for clarity.

Algorithms and Programming	
Educational Standard	**How Course Meets Standard**
2-AP-10 Use flowcharts and/or pseudocode to address complex problems as algorithms.	Students will learn how to create and use flowcharts in real-world situations as well as in their programming.
2-AP-11 Create clearly named variables that represent different data types and perform operations on their values.	Through coding and unplugged activities, students will learn what variables are and how to utilize them in their programs.
2-AP-12 Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.	All coding and unplugged activities introduce and incorporate these foundational skills. They are built upon and used throughout the different activities.
2-AP-13 Decompose problems and subproblems into parts to facilitate the design, implementation, and review of programs.	Students will look at and design flowcharts for real-world situations and their code to break down challenges into smaller pieces.
2-AP-14 Create procedures with parameters to organize code and make it easier to reuse.	Students will learn how to write and incorporate functions and parameters into their code. All coding activities require the use of functions and parameters.
2-AP-15 Seek and incorporate feedback from team members and users to refine a solution that meets user needs.	Collaboration will present students with the opportunity to receive other perspectives on their programs. In each coding activity, students are encouraged to ask others and use any additional resources to help fix any bugs or work through any challenges.
2-AP-16 Incorporate existing code, media, and libraries into original programs, and give attribution.	Coding activities utilize different platforms, such as Scratch and Godot, to modify, remix, and incorporate parts of existing programs to make them their own.

2-AP-17 Systematically test and refine programs using a range of test cases.	Students will actively practice a variety of strategies to fix and debug their code in every stage of the programming process.
2-AP-19 Document programs in order to make them easier to follow, test, and debug.	As students model how computers receive instructions, they will learn the importance of documentation in order to refine it in the future.

Impacts of Computing:	
Educational Standard	**How Class Meets Standard**
2-IC-20 Compare tradeoffs associated with computing technologies that affect people's everyday activities and career options.	Through the discussion of real-world situations, students will brainstorm ways to make smart and appropriate choices with technology. They will focus on how to stay safe and the anticipated results of the choices that they could make.
2-IC-21 Discuss issues of bias and accessibility in the design of existing technologies.	Students will discuss websites that present a certain angle in order to get a certain result. They will focus on how to keep themselves safe and informed to protect themselves from these unreliable sources.
2-IC-22 Collaborate with many contributors through strategies such as crowdsourcing or surveys when creating a computational artifact.	All coding activities require students to be open to and actively collaborate with their peers in order to troubleshoot and expand on their programs. All coding activities can additionally be completed in groups.
2-IC-23 Describe tradeoffs between allowing information to be public and keeping information private and secure.	Students will discuss the importance of digital security. They will discuss why, how, and when to share posts or keep them private.