



CodeHS

Computing Ideas (Lovelace) K-12 CS Framework Alignment Overview

The Computing Ideas course is a first computer science course introducing the basics of programming with Karel the Dog, the basics of designing a web page, how information and images are represented with computers, and the functionality and impact of the Internet. This document is an overview of how the Computing Ideas course aligns with the K-12 CS Framework's concepts and practices.

K-12 CS Framework Practices

Fostering an Inclusive Computing Culture

Through collaborative projects, students learn to include a diverse set of perspectives in the development process and functionality of their computational artifacts.

Collaborating Around Computing

This course provides several opportunities for students to create computational artifacts both individually and in teams. Several small group and full class activities are included in the curriculum. CodeHS provides guidance for implementing pair programming in the classroom.

Recognizing and Defining Computational Problems

As the problems in the course get more and more complex, the course teaches students to break down large, complex problems into manageable subproblems that can be solved independently. The course continually emphasizes decomposition and top down design.

Developing and Using Abstractions

The Computing Ideas course teaches students to develop abstractions to manage the complexity of their programs. Students develop and use their own abstractions to generalize their solutions and simplify the development process. Students learn about both procedural abstraction and data abstraction in this course. Students develop their own data encodings to represent high level ideas such as text and images as bits.

Creating Computational Artifacts

This course has a huge emphasis on creating computational artifacts. In each lesson of the course, students develop their own computational artifacts both for creative expression as well as to solve computational problems. Students create programs, digital images, websites, and digital presentations in the Computing Ideas course.

Testing and Refining Computational Artifacts

Testing and refining computational artifacts is an important part of the development process that is emphasized in the Computing Ideas course. The course teaches students to test their solutions, identify and fix errors, use debugging tools, and consider all edge cases when developing and testing their programs.

Communicating About Computing

This course gives students several opportunities to communicate their ideas and solutions to others. Students are encouraged to describe and justify their solutions to their teachers and their peers. Students are also required to document their code to communicate how it works. There are multiple research projects in this course that require students to research a computing topic of their choice and communicate their research through multiple forms of media.

K-12 CS Framework Concepts

Computing Systems

Computing Systems is a core concept throughout the Computing Ideas course. Students learn about various computing devices and how humans interact with them, including devices that extend the capabilities of humans. Students learn about computer organization including the relationship between hardware and software. Troubleshooting computing systems is a core concept of the Computing Ideas course as well. Computing systems might not work as expected because of problems in the software. Students are expected to identify problems in their programs and fix them.

Networks and the Internet

In The Internet unit, students learn about network communication and organization, Internet protocols, Internet addressing, the benefits of the packet-switched architecture of the Internet, and the effect of hierarchy and redundancy on the scalability and reliability of the Internet. Students learn about the importance of cybersecurity and the various security measures we take to protect information and privacy on the Internet.

Data and Analysis

The Computing Ideas course teaches students how data is stored in a computer as an abstract representation. Students learn exactly how text and image data is organized and stored as physical bits in a computing system. Students also learn how sensors must convert physical data into a digital representation that can be stored in a computer, and how data collection can be automated with sensors.

Algorithms and Programming

Algorithms and programming are central to the Computing Ideas course. Students learn the core principles of developing their own algorithms and implementing them in the JavaScript programming language. Algorithms, variables, control, modularity, and program development are all taught in this course.

Impacts of Computing

Computing has had significant impacts in several fields. In this course, students learn about the positive and negative impacts computing innovations and the Internet have had on culture, social interactions, safety, and privacy. Students also learn the ethical considerations of sharing their code with others, and finding solutions to CodeHS exercises online.