

Introduction

In order for students to become “computational thinkers” they need experience solving a wide range of problems and the opportunity to experiment with a variety of solution strategies. This unit begins with an introduction to the problem solving process. Students are asked to solve new problems by planning a strategy, designing and producing solutions, and then reflecting on their solutions and strategies.

Throughout the unit the emphasis should be on the process rather than the solution. Most of the world’s problems today do not have single simple solutions. In order to contribute effectively to the solution of these problems, students need to be comfortable in a collaborative environment where multiple approaches are valued and encouraged and where failure is seen as part of the process toward solution. Students must learn to think abstractly and apply known algorithms where appropriate, but also create new algorithms that can be applied to complex problems.

As students reflect on their solution processes and solutions and share those reflections with their peers, it is an opportunity to pull out instances where one strategy might be preferred over another and problems for which there are “standard” solutions versus those where there are many possible solutions.

Many of the problems presented have a mathematical basis and can serve to provide connections between mathematics and computer science. Common computer science topics such as searching, sorting, and graphing are introduced. Although programming the solutions to many of these problems is beyond the scope of this course, students will gain a basic understanding of the algorithms and be able to analyze them. In particular, it is important to emphasize that the models used for solving computational problems are the underpinnings of computer science and as such remain largely the same even as we add new tools and languages.

A key point of emphasis throughout the unit is the connection between the solution “process” and the discussion toward the end of Unit 1 related to how computers are programmed. It is also important to emphasize that not all problems are easily solved by computers.

Specific topics for each instructional day are listed in the overview chart on the next page.

Daily Overview Chart	
Instructional Day	Topic
1-2	Introduce data collection and problem solving.
3	Introduce the four steps of the problem solving process.
4-6	Apply the problem solving process. Use different strategies to plan and carry out the plan to solve several problems.
7-9	Reinforce the four steps of the problems solving process.
10-12	Count in the binary number system. Convert between binary and decimal numbers in the context of topics that are important to computer science.
13-14	Introduce the linear and binary search algorithms.
15-16	Explore sorted and unsorted lists and various sorting algorithms.
17	Introduce minimal spanning trees and how graphs can be used to help solve problems.
18-21	Final projects and presentations