# Readington Township Public Schools

# Coding 1
# (7th grade and 1st-year 8th-grade students)

**Authored by:** Lora Petersen

**Reviewed by:** Sarah Pauch
Supervisor of Math, Science, and Technology

**Approval Date:** September 13, 2022

## I. OVERVIEW

Coding 1 is the 2nd of 3 cycle courses in Computer Science offered in RMS. New approaches necessary for solving the critical challenges that we face as a society will require harnessing the power of technology and computing. Rapidly changing technologies and the proliferation of digital information have permeated and radically transformed learning, working, and everyday life. To be well-educated, global-minded individuals in a computing-intensive world, students must have a clear understanding of the concepts and practices of computer science. As education systems adapt to a vision of students who are not just computer users but also computationally literate creators who are proficient in the concepts and practices of computer science and design thinking, engaging students in computational thinking and human-centered approaches to design through the study of computer science and technology serves to prepare students to ethically produce and critically consume technology. (New Jersey Department of Education)

## II. STUDENT OUTCOMES (2020 New Jersey Student Learning Standards – Computer Science)

The course objectives will cover but are not limited to these standards:

**Computing Systems**

8.1.8.CS.1: Recommend improvements to computing devices in order to improve the ways users interact with the devices.

8.1.8.CS.3: Justify design decisions and explain potential system trade-offs.

8.1.8.CS.4: Systematically apply troubleshooting strategies to identify and resolve hardware and software problems in computing systems.

**Impacts of Computing**

8.1.8.IC.1: Compare the trade-offs associated with computing technologies that affect individual's everyday activities and career options.

8.1.8.IC.2: Describe issues of bias and accessibility in the design of existing technologies.

**Algorithms & Programming**

8.1.8.AP.1: Design and illustrate algorithms that solve complex problems using flowcharts and/or pseudocode.

8.1.8.AP.2: Create clearly named variables that represent different data types and perform operations on their values.

8.1.8.AP.3: Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals.

8.1.8.AP.4: Decompose problems and sub-problems into parts to facilitate the design, implementation, and review of programs.

8.1.8.AP.5: Create procedures with parameters to organize code and make it easier to reuse.

8.1.8.AP.6: Refine a solution that meets users' needs by incorporating feedback from team members and users.

8.1.8.AP.7: Design programs, incorporating existing code, media, and libraries, and give attribution.

8.1.8.AP.8: Systematically test and refine programs using a range of test cases and users.

8.1.8.AP.9: Document programs in order to make them easier to follow, test, and debug.

## III. COURSE OBJECTIVES

### Foundations of Coding

- Introduction
  Students will be able to:
  - Become familiar with the coding environment used in class
  - Order movement commands as sequential steps in a program

- Events
  Students will be able to:
  - Define an event
  - Identify actions that correlate to input events
  - Use a variety of events to trigger scripts
  - Create interactive projects using sequence and event-handlers
  - Share creative artifacts with other students
- Loops
  Students will be able to:
  - Break complex tasks into smaller repeatable sections
  - Recognize large repeated patterns as made from smaller repeated patterns
  - Identify the benefits of using a loop structure instead of manual repetition
  - Distinguish between loops that repeat a fixed number of times and loops that repeat as long as a condition is true
- Debugging Programs
  Students will be able to:
  - Identify a bug and the problems it causes in a program
  - Describe and implement a plan to debug a program
  - Modify an existing program to solve errors
- Conditionals
  Students will be able to:
  - Translate spoken language conditional statements and loops into a program
  - Create projects using a combination of looped sequences and conditionals
  - Build programs with the understanding of multiple strategies to implement conditionals
- Decomposition
  Students will be able to:
  - Decompose a series of events
  - Create scripts that will trigger the action of one sprite dependent on the action of another sprite
  - Plan and create an animation based on a set of events and actions
  - Identify the actions that occur based on certain events
- **Design and Concept Development**
  Students will be able to follow the steps of the design process when working on their projects: (the process may look similar to this):
  - Ask: Identify the Need and Constraints
  - Research the Problem
  - Imagine: Develop Possible Solutions
  - Plan: Select a Promising Solution
  - Create: Build a Prototype
  - Test and Evaluate Prototype
  - Improve: Redesign as Needed

## IV. STRATEGIES

Strategies  may include but are not limited to:

- Group discussions
- Teacher presentation
- Student projects
- Guided groups
- One to one instruction
- Interactive SmartBoard lessons
- Tutorials
- Online practice

## V. EVALUATION

Assessments may include but are not limited to:

- Teacher Observations
- Class Participation
- Class Discussions
- Class Assignments
- Homework Assignments
- Notebooks
- Student Projects
- Tests and Quizzes
- Anecdotal Records
- Presentations

## VI. REQUIRED RESOURCES

- Scratch
- Scratch Encore
- Chromebooks
- Headphones

Supplemental Resources may include, but are not limited to:

- Code Monkey
- Code.org
- Code Master
- Tynker
- Makey Makey Classic
- Ozobots
- Code Studios Lesson
- CS First
- Beanz Magazine
- Common Sense Media
- Kahoot
- Blooket
- BrainPop
- Khan Academy

## VII. SCOPE AND SEQUENCE

- Foundations of Coding (20 days)

  - Introduction
  - Events
  - Loops
  - Debugging Process
  - Conditionals
  - Decomposition

- Design and Concept Development (20 days)
  - Design Process
  - Coding projects in various languages (such as “blockly”)
  - Design, Produce, and Share Various Projects