

# Unit 03: Computer Science Concepts

Content Area: **Template**  
Course(s):  
Time Period: **Full Year**  
Length: **FY**  
Status: **Published**

## Standards Alignment

---

### New Jersey Student Learning Standards

---

#### **Practice 1. Asking questions (for science) and defining problems (for engineering)**

**Asking questions and defining problems in 6–8 builds on K–5 experiences and progresses to specifying relationships between variables, and clarifying arguments and models.**

Ask questions that arise from careful observation of phenomena, models, or unexpected results, to clarify and/or seek additional information.

Ask questions to determine relationships between independent and dependent variables and relationships in models.

Ask questions to clarify and/or refine a model, an explanation, or an engineering problem.

Ask questions that require sufficient and appropriate empirical evidence to answer.

Ask questions that challenge the premise(s) of an argument or the interpretation of a data set.

Define a design problem that can be solved through the development of an object, tool, process or system and includes multiple criteria and constraints, including scientific knowledge that may limit possible solutions.

#### **Practice 2. Developing and using models**

**Modeling in 6–8 builds on K–5 experiences and progresses to developing, using, and revising models to describe, test, and predict more abstract phenomena and design systems.**

Evaluate limitations of a model for a proposed object or tool.

Develop or modify a model—based on evidence – to match what happens if a variable or component of a system is changed.

Develop and/or revise a model to show the relationships among variables, including those that are not observable but predict observable phenomena.

Develop a model to describe unobservable mechanisms.

Develop and/or use a model to generate data to test ideas about phenomena in natural or designed systems, including those representing inputs and outputs, and those at unobservable scales.

#### **Practice 5. Using mathematics and computational thinking**

**Mathematical and computational thinking in 6–8 builds on K–5 experiences and progresses to identifying patterns in large data sets and using mathematical concepts to support explanations and arguments.**

Use digital tools (e.g., computers) to analyze very large data sets for patterns and trends.

Use mathematical representations to describe and/or support scientific conclusions and design solutions.

Create algorithms (a series of ordered steps) to solve a problem.

Apply mathematical concepts and/or processes (e.g., ratio, rate, percent, basic operations, simple algebra) to scientific and engineering questions and problems.

Use digital tools and/or mathematical concepts and arguments to test and compare proposed solutions to an engineering design problem.

**Practice 6. Constructing explanations (for science) and designing solutions (for engineering)**

**Constructing explanations and designing solutions in 6–8 builds on K–5 experiences and progresses to include constructing explanations and designing solutions supported by multiple sources of evidence consistent with scientific ideas, principles, and theories.**

Construct an explanation using models or representations.

Apply scientific reasoning to show why the data or evidence is adequate for the explanation or conclusion.

Apply scientific ideas or principles to design, construct, and/or test a design of an object, tool, process or system.

Undertake a design project, engaging in the design cycle, to construct and/or implement a solution that meets specific design criteria and constraints.

Optimize performance of a design by prioritizing criteria, making tradeoffs, testing, revising, and re-testing.

**Practice 8. Obtaining, evaluating, and communicating information**

**Obtaining, evaluating, and communicating information in 6–8 builds on K–5 experiences and progresses to evaluating the merit and validity of ideas and methods.**

Communicate scientific and/or technical information (e.g. about a proposed object, tool, process, system) in writing and/or through oral presentations.

**Crosscutting Statements**

**4. Systems and System Models – A system is an organized group of related objects or components; models can be used for understanding and predicting the behavior of systems.**

Systems may interact with other systems; they may have sub-systems and be a part of larger complex systems.

Models can be used to represent systems and their interactions—such as inputs, processes and outputs—and energy, matter, and information flows within systems.

---

## **Integration of Career Readiness, Life Literacies and Key Skills**

CRP.K-12.CRP1	Act as a responsible and contributing citizen and employee.
CRP.K-12.CRP2	Apply appropriate academic and technical skills.
CRP.K-12.CRP3	Attend to personal health and financial well-being.
CRP.K-12.CRP4	Communicate clearly and effectively and with reason.
CRP.K-12.CRP5	Consider the environmental, social and economic impacts of decisions.
CRP.K-12.CRP6	Demonstrate creativity and innovation.
CRP.K-12.CRP7	Employ valid and reliable research strategies.
CRP.K-12.CRP8	Utilize critical thinking to make sense of problems and persevere in solving them.
CRP.K-12.CRP9	Model integrity, ethical leadership and effective management.
CRP.K-12.CRP10	Plan education and career paths aligned to personal goals.
CRP.K-12.CRP11	Use technology to enhance productivity.
CRP.K-12.CRP12	Work productively in teams while using cultural global competence.

## **Technology / Integration of Computer Science and Design Thinking**

---

TECH.8.1.8	Educational Technology: All students will use digital tools to access, manage, evaluate, and synthesize information in order to solve problems individually and collaborate and to create and communicate knowledge.
TECH.8.1.8.A	Technology Operations and Concepts: Students demonstrate a sound understanding of technology concepts, systems and operations.
TECH.8.1.8.A.1	Demonstrate knowledge of a real world problem using digital tools.
TECH.8.2.8	Technology Education, Engineering, Design, and Computational Thinking - Programming: All students will develop an understanding of the nature and impact of technology, engineering, technological design, computational thinking and the designed world as they relate to the individual, global society, and the environment.
TECH.8.2.8.A	The Nature of Technology: Creativity and Innovation: Technology systems impact every aspect of the world in which we live.
TECH.8.2.8.A.2	Examine a system, consider how each part relates to other parts, and discuss a part to redesign to improve the system.
TECH.8.2.8.A.3	Investigate a malfunction in any part of a system and identify its impacts.
TECH.8.2.8.B	Technology and Society: Knowledge and understanding of human, cultural and society values are fundamental when designing technology systems and products in the global society.
TECH.8.2.8.B.2	Identify the desired and undesired consequences from the use of a product or system.
TECH.8.2.8.C	Design: The design process is a systematic approach to solving problems.
TECH.8.2.8.C.2	Explain the need for optimization in a design process.
TECH.8.2.8.C.3	Evaluate the function, value, and aesthetics of a technological product or system, from the perspective of the user and the producer.
TECH.8.2.8.C.5b	Create a technical sketch of a product with materials and measurements labeled.
TECH.8.2.8.E	Computational Thinking: Programming: Computational thinking builds and enhances problem solving, allowing students to move beyond using knowledge to creating knowledge.
TECH.8.2.8.E.2	Demonstrate an understanding of the relationship between hardware and software.
TECH.8.2.8.E.3	Develop an algorithm to solve an assigned problem using a specified set of commands and use peer review to critique the solution.
TECH.8.2.8.E.4	Use appropriate terms in conversation (e.g., programming, language, data, RAM, ROM, Boolean logic terms).

## **Interdisciplinary Connections: NJSLs for ELA, Social Studies, Science and/or Math Section**

---

LA.K-12.NJLSA.W	Writing
LA.K-12.NJLSA.W2	Write informative/explanatory texts to examine and convey complex ideas and information clearly and accurately through the effective selection, organization, and analysis of content.
LA.W.8.2	Write informative/explanatory texts to examine a topic and convey ideas, concepts, and information through the selection, organization, and analysis of relevant content.
LA.W.8.2.D	Use precise language and domain-specific vocabulary to inform about or explain the topic.

## **Integration of Diversity, Equity and Inclusion; Climate Change; Informational and Media Literacy**

---

see Crosswalks

## **21st Century Life and Careers**

---

### **Stage I: Desired Results**

---

### **Transfer/Overview/Rationale**

<b>Transfer / Overview / Rationale</b>
<p>Unit Rationale The purpose of this unit...</p> <p><b>Iteration, Procedures, Functions, and Advanced Concepts</b></p> <p>In this unit, students will further apply the aspects of computational thinking in the performance of more advanced programming concepts. Iteration will introduce loops and repeated states. Procedures and Functions will introduce recursion of smaller programs into larger ones to increase efficiency and promote good coding practice. Other advanced concepts in coding will introduce nesting, data storage, sensor input, internet access, and modularity. By learning these concepts, students will be better equipped to create their own programs.</p>

### **Meaning**

---

### **Essential Questions**

---

Essential Questions

How is computational thinking used in the performance of iteration, procedures, and functions within code?

How and why are loops used in coding?

What are functions and procedures? How are they similar and how do they differ?

Why are nesting, storage, sensors, and modularity important to computer programming today?

## **Enduring Understanding/Indicators of Understanding**

---

Enduring Understanding/Indicators of Understanding

Computational thinking involves abstraction, algorithms, decomposition, pattern recognition, and generalization.

Iteration is the repetition of a block of code to reach a desired output.

Procedures and Functions are employed to minimize the writing of new blocks of code to accomplish similar tasks to those already coded.

Coding nesting, data storage, sensors, and modularity help to create a more robust computer program.

## **Acquisition (Student Learning Objectives)**

---

### **Knowledge**

---

Knowledge

Students will know...

Using computational thinking allows us to see a program as a convergence of smaller systems into a larger one.

Iteration is the creation of loops of code to repeat a process over and over until a condition is met.

Procedures are smaller programs that can receive input and return output that we write to perform an action as often as needed within a larger program.

Functions are procedures that are already built-in to a programming language to make coding common things easier and more efficient.

Advanced coding concepts are used to create more robust programs that are able to perform more tasks more efficiently.

## **Skills**

---

Skills

Student will be skilled at ...

Combining multiple coding concepts to create more advanced programs.

Developing efficient answers to more complicated coding problems using iteration, functions, procedures, and other advanced concepts.

Recognizing the proper coding technique/s to use to create better code.

## **Stage 3: Learning Plan**

---

## Resource and Mentor Texts

---

### Resources and Mentor Texts

MIT's AppInventor - <http://appinventor.mit.edu/explore/>

App Inventor EDU - <https://sites.google.com/site/appinventoredu/curriculum>

Google's Computational Thinking for Educators -  
[https://computationalthinkingcourse.withgoogle.com/course?use\\_last\\_location=true](https://computationalthinkingcourse.withgoogle.com/course?use_last_location=true)

Google CS-First - <https://csfirst.withgoogle.com/en/home>

K12 Computer Science Framework - <https://k12cs.org/wp-content/uploads/2016/09/K%E2%80%9312-Computer-Science-Framework.pdf>

Technopedia - <https://www.techopedia.com/>

AppInventor Code Snippet Cards - <http://explore.appinventor.mit.edu/resources/app-inventor-flash-cards>

"I Love my Smartphone: Mobile App Development V2" -  
[http://www.royalsoced.org.uk/1035\\_MobileAppDevelopment.html](http://www.royalsoced.org.uk/1035_MobileAppDevelopment.html)

"Android Game Development with AppInventor" by Anshul Bhagi -  
[http://explore.appinventor.mit.edu/sites/all/files/Resources/Thesis\\_FINAL\\_AnshulBhagi.pdf](http://explore.appinventor.mit.edu/sites/all/files/Resources/Thesis_FINAL_AnshulBhagi.pdf)

"MIT App Inventor: Enabling Personal Mobile Computing" - Shaileen Pokress, Jose Veiga -  
<https://arxiv.org/pdf/1310.2830.pdf>

The App Inventor Course-in-a-box - <http://www.appinventor.org/content/CourseInABox/Intro/courseinabox>

Middle School Android - <https://sites.google.com/site/middleschoolandroid/home/app-inventor-tutorials>

CS128 Introduction to Software Application Development Fall 2012 - <https://sites.google.com/a/jcu.edu/cs128fall2012/home>

“Python Methodologies for Data Sciences” by Lars Sorenson - <https://www.cs.rutgers.edu/~biglars/PMDS/>

## **Formative Assessment Strategies**

---

Formative Assessment Strategies

Teacher observation

Class discussion

Coding example activities

Google Classroom Questions

Check-ins for original program assignment

## **Learning Activities/Unit of Study**

---

Learning Activities/Unit of Study

Google Classroom Daily Questions

ApplInventor follow-along coding activities, with opportunity for extension

Computational thinking discussion and activities with advanced concepts

Code construction and deconstruction activities

Introduction of final project coding activity

## **Modifications and/or Accommodations**

---

### **Suggested Modifications (ELL, Sp. Ed, Gifted, At-risk of Failure)**

#### **English Language Learners**

**Native language support:** The teacher provides auditory or written content to students in their native language.

**Adjusted Speech:** The teacher changes speech patterns to increase student comprehension. This could include facing the students, paraphrasing, clearly indicating the most important ideas, and speaking more slowly.

**Visuals:** The teacher uses graphics, pictures, visuals, and manipulatives. This helps ELL students better understand and comprehend the subjects at hand.

**Front-Loading Vocabulary:** The teacher front loads vocabulary. This means providing students with a list of important vocabulary words they will need to know for a book, lesson, etc. prior to the lesson being taught. Including pictures to go with the vocabulary words is also very beneficial for the students.

#### **Special Education Students**

**Chunking:** The teacher presents information in a way that makes it easy for students to understand and remember. Chunking is based on the presumption that our working memory is easily overloaded by excessive detail. The best way to deliver information is to organize it into meaningful units. Because students with special needs get overloaded easily, chunking is an effective strategy to use with them.

**Checking for Understanding:** It is important to constantly check for understanding, especially for students who have accommodations. Teachers want to make sure students understand the concepts being covered in a way that makes sense to them.

**Extra time:** The teacher provides students with special needs extra time to complete work or answer

questions. It is important to give students enough time to process their thoughts.

**Oral Reading:** The teacher will read work orally to students. Class work such as tests and literature circles may need to be read aloud to the student.

**Timers:** The teacher will use timers as an instructional tool. The use of timers is beneficial for students who have trouble completing tasks. Timers can be helpful so the student is aware of how much time they have to complete an assignment.

## **Students with 504 Plans**

**Chunking:** The teacher presents information in a way that makes it easy for students to understand and remember. Chunking is based on the presumption that our working memory is easily overloaded by excessive detail. The best way to deliver information is to organize it into meaningful units. Because students with special needs get overloaded easily, chunking is an effective strategy to use with them.

**Checking for Understanding:** It is important to constantly check for understanding, especially for students who have accommodations. Teachers want to make sure students understand the concepts being covered in a way that makes sense to them.

**Extra time:** The teacher provides students with special needs extra time to complete work or answer questions. It is important to give students enough time to process their thoughts.

## **Gifted & Talented Strategies**

**Extensions/Enrichments:** Teachers will provide gifted and talented students with extension/enrichment projects. Students will be challenged to further their understanding, to apply acquired knowledge, and/or to produce something in reference to acquired knowledge.

**Modify/Change Activities:** Teachers will monitor and modify activities to accommodate those students who need to be challenged further. Additional reading, problem-solving, writing, or project work is necessary for those students who are ready to move on at a rate more accelerated than their peers. In this way, G & T students are provided the same opportunity for support as special needs students.

## **Students at Risk of School Failure**

**Directions or Instructions:** Make sure directions and/or instructions are given in limited numbers. Give directions/instructions verbally and in simple written format. Ask students to repeat the instructions or directions to ensure understanding occurs. Check back with the student to ensure he/she hasn't forgotten.

**Peer Support:** Peers can help build confidence in other students by assisting in peer learning. Many teachers use the 'ask 3 before me' approach. This is fine, however, a student at risk may have to

have a specific student or two to ask. Set this up for the student so he/she knows who to ask for clarification before going to you.

**Alternate or Modified Assignments:** Always ask yourself, "How can I modify this assignment to ensure the students at risk are able to complete it?" Sometimes you'll simplify the task, reduce the length of the assignment or allow for a different mode of delivery. For instance, many students may hand something in, the at-risk student may jot notes and give you the information verbally. Or, it just may be that you will need to assign an alternate assignment.

**Increase One to One Time:** When other students are working, always touch base with your students at risk and find out if they're on track or needing some additional support. A few minutes here and there will go a long way to intervene as the need presents itself.

**Contracts:** It helps to have a working contract between you and your students at risk. This helps prioritize the tasks that need to be done and ensure completion happens. Each day write down what needs to be completed, as the tasks are done, provide a checkmark or happy face. The goal of using contracts is to eventually have the student come to you for completion sign-offs.

**Hands On:** As much as possible, think in concrete terms and provide hands-on tasks. This means a child doing math may require a calculator or counters. The child may need to tape record comprehension activities instead of writing them. A child may have to listen to a story being read instead of reading it him/herself.

**Tests/Assessments:** Tests can be done orally if need be. Break tests down in smaller increments by having a portion of the test in the morning, another portion after lunch and the final part the next day.

**Seating:** Seat students near a helping peer or with quick access to the teacher. Those with hearing or sight issues need to be close to the instruction which often means near the front.