

# Unit 10: Recursion

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

## UNIT RATIONALE

---

Sometimes a problem can be solved by solving smaller or simpler versions of the same problem rather than attempting an iterative solution. This is called recursion, and it is a powerful math and computer science idea. In this unit, students will revisit how control is passed when methods are called, which is necessary knowledge when working with recursion. Tracing skills introduced in Unit 2 are helpful for determining the purpose or output of a recursive method. In this unit, students will learn how to write simple recursive methods and determine the purpose or output of a recursive method by tracing.

## ESSENTIAL QUESTIONS

---

What real-world processes do you follow that are recursive in nature?

Why do programmers sometimes prefer using recursive solutions when sorting data in a large data set?

## STANDARDS

---

### NEW JERSEY STUDENT LEARNING STANDARDS: CONTENT AREA

---

#### New Jersey (NJSL) - Grades 9-12 - Computer Science and Design Thinking (2020)

##### 8.1.12.AP.1:

Design algorithms to solve computational problems using a combination of original and existing algorithms.

##### 8.1.12.AP.2:

Create generalized computational solutions using collections instead of repeatedly using simple variables.

##### 8.1.12.AP.3:

Select and combine control structures for a specific application based upon performance and readability, and identify trade-offs to justify the choice.

##### 8.1.12.AP.4:

Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue.

##### 8.1.12.AP.5:

Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.

### 8.1.12.AP.6:

Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.

CS.9-12.8.1.12.AP.1	Design algorithms to solve computational problems using a combination of original and existing algorithms.
CS.9-12.8.1.12.AP.2	Create generalized computational solutions using collections instead of repeatedly using simple variables.
CS.9-12.8.1.12.AP.3	Select and combine control structures for a specific application based upon performance and readability, and identify trade-offs to justify the choice.
CS.9-12.8.1.12.AP.4	Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue.
CS.9-12.8.1.12.AP.5	Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.
CS.9-12.8.1.12.AP.6	Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.

## **NEW JERSEY STUDENT LEARNING STANDARDS: CAREER READINESS, LIFE LITERACIES AND KEY SKILLS**

---

TECH.9.4.12.CT.1	Identify problem-solving strategies used in the development of an innovative product or practice (e.g., 1.1.12acc.C1b, 2.2.12.PF.3).
TECH.9.4.12.CT.2	Explain the potential benefits of collaborating to enhance critical thinking and problem solving (e.g., 1.3E.12profCR3.a).

## **NEW JERSEY STUDENT LEARNING STANDARDS: COMPUTER SCIENCE AND DESIGN THINKING**

---

See content area standards.

## **PRE-ASSESSMENTS**

---

Discussion Questions:

- What is the difference between for loops and while loops?
- What is a nested loop?

Consider the following code segment:

```
int rows = 5;
for(int i = 1; i <= rows; i++)
{
    for(int j = 5; j >= i; j--)
    {
        System.out.print(j + " ");
    }
    System.out.println();
}
```

}

- How many numbers will be printed in this program?

## INSTRUCTIONAL PLAN

---

### MODULE 10

---

#### LESSON 10.1

---

##### Lesson 10.1: Recursion

In this lesson, students will learn about recursion and recursive methods. A recursive method is a method that calls itself. Recursive methods contain at least one base case, which halts the recursion, and at least one recursive call. This lesson corresponds with AP Computer Science A topic 10.1.

<b>Student Learning Intentions (SLI) WALT: (We are learning to...)</b>	<ul style="list-style-type: none"><li>• Determine the result of executing recursive methods</li></ul>
<b>Student Learning Strategies</b>	Code tracing Create a plan Error analysis Identify a subtask Look for a pattern Pair programming Predict and compare Simplify the problem Think aloud
<b>Success Criteria</b>	Check for understanding Completion of below activities AP Topic Questions
<b>Formative Assessment (drives instructional decisions)</b>	Teacher Observation Check for Understanding AP Topic Questions

## Activities and Resources

- Watch the lesson video and take the corresponding quiz. This quiz is a quick check for understanding.
- Explore the Summing example.
- Explore the Sum Array example.
- Explore the Fibonacci Recursion example.
- Complete the Factorial exercise.
- Complete the Countdown! exercise.
- Complete the Recursive Minimum exercise.
- Complete the Bacteria Cultures exercise.
  - This is a culminating activity that can be used as an individual or small group project/lab.
- Complete the Mapping Out Recursion handout.

## Suggested Modifications

### English Language Learners

**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.

### Students with Individualized Education Plans/504s

**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.

### **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.

---

## LESSON 10.2

### Lesson 10.2: Recursive Searching

In this lesson, students will examine and apply recursive searching such as the binary search. The binary search algorithm starts at the middle of a sorted array or ArrayList and eliminates half of the array or ArrayList in each iteration until the desired value is found or all elements have been eliminated. This lesson corresponds with AP Computer Science A topic 10.2.

**Student Learning Intentions (SLI) WALT: (We are learning to...)**

- Apply recursive search algorithms to information in String, 1D array, or ArrayList objects

<b>Student Learning Strategies</b>	<p>Code tracing</p> <p>Create a plan</p> <p>Error analysis</p> <p>Identify a subtask</p> <p>Look for a pattern</p> <p>Pair programming</p> <p>Predict and compare</p> <p>Simplify the problem</p> <p>Think aloud</p>
<b>Success Criteria</b>	<p>Check for understanding</p> <p>Completion of below activities</p> <p>AP Topic Questions</p>
<b>Formative Assessment (drives instructional decisions)</b>	<p>Teacher Observation</p> <p>Check for Understanding</p> <p>AP Topic Questions</p>
<b>Activities and Resources</b>	<ul style="list-style-type: none"> <li>• Watch the lesson video and take the corresponding quiz. This quiz is a quick check for understanding.</li> <li>• Explore the Binary Search example.</li> <li>• Explore the Binary Recursive example.</li> <li>• Explore the Binary vs Linear example.</li> <li>• Complete the Exploring Binary Searches exercise.</li> <li>• Complete the Comparing Binary Search and Linear Search exercise.</li> <li>• Complete the Maximum Iterations exercise.</li> <li>• Complete the Binary vs Linear Search handout.</li> </ul>
<b>Suggested Modifications</b>	<p>See Lesson 10.1 above.</p>

## LESSON 10.3

### Lesson 10.3: Recursive Sorting

Apply recursive algorithms to sort elements of an array or ArrayList objects.

<b>Student Learning Intentions (SLI) WALT: (We are learning to...)</b>	<ul style="list-style-type: none"> <li>• Apply recursive search algorithms to information in String, 1D array, or ArrayList objects</li> </ul>
<b>Student Learning Strategies</b>	<p>Code tracing</p>

	<p>Create a plan</p> <p>Error analysis</p> <p>Identify a subtask</p> <p>Look for a pattern</p> <p>Pair programming</p> <p>Predict and compare</p> <p>Simplify the problem</p> <p>Think aloud</p>
<b>Success Criteria</b>	<p>Check for understanding</p> <p>Completion of below activities</p> <p>AP Topic Questions</p>
<b>Formative Assessment (drives instructional decisions)</b>	<p>Teacher Observation</p> <p>Check for Understanding</p> <p>AP Topic Questions</p>
<b>Activities and Resources</b>	<ul style="list-style-type: none"> <li>• Watch the lesson video and take the corresponding quiz. This quiz is a quick check for understanding.</li> <li>• Explore the Merge Sort example.</li> <li>• Explore the Sort Visualizer example.</li> <li>• Complete the Explore Merge Sort exercise.</li> <li>• Complete the Merge Sort Benchmark Testing exercise.</li> <li>• Complete the Recursive Calls exercise.</li> <li>• Complete the Sort Benchmark Testing exercise.</li> </ul>
<b>Suggested Modifications</b>	<p>See Lesson 10.1 above.</p>

## REFLECTIONS

---

### INTERDISCIPLINARY CONNECTIONS: NEW JERSEY STUDENT LEARNING STANDARDS FOR ELA, SOCIAL STUDIES, SCIENCE AND/OR MATHEMATICS

---

#### CCSS - English-Language Arts

##### Key Ideas and Details:

CCSS.ELA-LITERACY.RL.11-12.1 Cite strong and thorough textual evidence to support analysis of what the text says explicitly as well as inferences drawn from the text, including determining where the text leaves matters uncertain.

##### Integration of Knowledge and Ideas:

CCSS.ELA-LITERACY.W.11-12.1 Write arguments to support claims in an analysis of substantive topics or texts, using valid reasoning and relevant and sufficient evidence.

Production and Distribution of Writing:

CCSS.ELA-LITERACY.W.11-12.4 Produce clear and coherent writing in which the development, organization, and style are appropriate to task, purpose, and audience.

Research to Build and Present Knowledge:

CCSS.ELA-LITERACY.W.11-12.7 Conduct short as well as more sustained research projects to answer a question (including a self-generated question) or solve a problem; narrow or broaden the inquiry when appropriate; synthesize multiple sources on the subject, demonstrating understanding of the subject under investigation.

Range of Writing:

CCSS.ELA-LITERACY.W.11-12.10 Write routinely over extended time frames (time for research, reflection, and revision) and shorter time frames (a single sitting or a day or two) for a range of tasks, purposes, and audiences

### **CCSS - Mathematics**

Reason quantitatively and use units to solve problems:

CCSS.MATH.CONTENT.HSN-Q.A.2 Define appropriate quantities for the purpose of descriptive modeling.

Create equations that describe numbers or relationships:

CCSS.MATH.CONTENT.HSA-CED.A.1 Create equations and inequalities in one variable and use them to solve problems.

Analyze functions using different representations:

CCSS.MATH.CONTENT.HSF-IF.C.7 Graph functions expressed symbolically and show key features of the graph.

Apply geometric concepts in modeling situations:

CCSS.MATH.CONTENT.HSG-MG.A.1 Use geometric shapes, their measures, and their properties to describe objects

Calculate expected values and use them to solve problems:

CCSS.MATH.CONTENT.HSS-MD.A.1 Define a random variable for a quantity of interest by assigning a numerical value to each event in a sample space;

CCSS.MATH.CONTENT.HSS-MD.A.2 Calculate the expected value of a random variable;

### **English Language Arts**

- Journal writing
- Close reading of industry-related content
- Create a brochure for a specific industry
- Keep a running word wall of industry vocabulary

### **Social Studies**

- Research the history of a given industry/profession
- Research prominent historical individuals in a given industry/profession
- Use historical references to solve problems

### **World Language**

- Translate industry-content
- Create a translated index of industry vocabulary
- Generate a translated list of words and phrases related to information technology

### **Math**

- Compare and contrast use of equations and variables in algebra and programming.
- Program graphics and use the properties of geometric shapes
- Compare the computer graphic coordinate system with the Cartesian coordinate plane in math

- Compare probability and the use of random numbers in computer programming.
- Track and track various data, such as industry's impact on the GDP, career opportunities or among of individuals currently occupying careers

### **Fine & Performing Arts**

- Create a poster recruiting young people to focus their studies on a career in Information Technology

### **Science**

- Research the environmental impact of a given career or industry
- Research latest developments in Information technology
- Investigate applicable-careers in STEM fields