

**Summit Public Schools**  
**Summit, New Jersey**  
**Grade Level / Content Area: Mathematics**  
**Length of Course: ½ Academic Year**  
**Curriculum: Introduction to Java Programming**  
**Updated: Spring 2019**

Developed By  
Brian Weinfeld

**Course Description:** This elective course serves as a direct follow up to the “Introduction to Computer Science” course and introduces students to the Java programming language. Over the course of the semester students will develop strong Java programming skills through a series of programming and written assignments. Students who complete this course with a B- or higher will be prepared to take an AP Computer Science course.

Topics that will be taught include a wide variety of techniques that include if/else, while, and for statements, variable creation and comparison and basic array use. They will have practical experience in problem solving skills. In addition, students will learn about the basics of object oriented programming.

## Anticipated Timetable for Intro to Java Programming

### QUARTER 3

#### Unit 0 – Introduction to Programming

Topic	Time Frame
Overview of IDE	1
Introduction activity	1
Practice Project #0	2
Total	4

#### Unit 1 – Custom Methods

Topic	Time Frame
Creating turnRight() method and practice custom methods	2
Practice programming activity	1
Project #1	3
Total	6

#### Unit 2 – Conditionals

Topic	Time Frame
Introduction to if statements and Boolean methods	2
Solving generic tasks	1
Project #2	3
Total	6

#### Unit 3 – Generic Code

Topic	Time Frame
-------	------------

Incorporating Boolean methods, conditional statements and custom methods.	2
Practice Activity	1
How to solve a complex problem. Breaking down complex topics.	1
Project #3	3
Total	7

#### Unit 4 – Methods and their Return Types

Topic	Time Frame
Extending Boolean methods to ints.	1
Using variables (Booleans and ints)	2
Project #4	4
Total	7

#### Unit 5 – Variables and Parameters

Topic	Time Frame
How to use variables and parameters	2
The similarities and difference between variables and parameters	2
Scope	1
Project #5	4
Total	9

#### Unit 6 – Theme 1 Final Project

Topic	Time Frame
Instance Variables	2
Project #6	6
Total	8

### QUARTER 4

#### Unit 7 – Class Creation

Topic	Time Frame
Class structure and skeleton	1
Constructors and instance variables	1
Class creation activity	2
Project #7	4
Total	8

### Unit 8 – Class Interaction and OOP Basics

Topic	Time Frame
How classes interact	1
Introduction to Object Oriented Programming	2
Encapsulation	1
Revisiting instance variables ('has-a')	1
Gets/sets and equals	2
Project #8	4
Total	11

### Unit 9 – Introduction to Arrays

Topic	Time Frame
Introduction to for loops	1
Introduction to arrays	2
Arrays practice	1
Project #9	5
Total	9

### Unit 10 – Introduction to Strings

Topic	Time Frame
Introduction to Strings	1
String methods	2
Strings vs Arrays (immutable)	1
Project #10	5
Total	9

### Unit 11 – Multiple Class Interactions and OOP Concepts

Topic	Time Frame
Revisiting writing a class (proper organization)	1
toString()	1
Multiple class interaction	2
Project #11	6
Total	10

### Unit 12 – Integration of Arrays, Strings and OOP Concepts

Topic	Time Frame
Revisiting string, arrays and class creation	2
Integrating all previous topics	2
Light introduction to pass by reference	1
Project #12	6

Total	11
-------	----

## Theme 1: Introduction to Java Programming

### Unit 0: Introduction to Programming

Standard	
<p><b>Big Ideas:</b> <i>Course Objectives / Content Statement(s)</i></p> <ul style="list-style-type: none"> <li>● Students will learn how to create and write their first program</li> <li>● Students will learn the 4 core Wombat methods (walk, turnLeft, pickLeaf, placeLeaf)</li> <li>● Students will learn how to properly format their code for readability</li> </ul>	
<p><b>Essential Questions</b></p> <p><i>What provocative questions will foster inquiry, understanding, and transfer of learning?</i></p>	<p><b>Enduring Understandings</b></p> <p><i>What will students understand about the big ideas?</i></p>
<ul style="list-style-type: none"> <li>● How is a line of Java code structured?</li> <li>● What is a syntax error?</li> <li>● What is the who/what relationship of each line of code?</li> </ul>	<p>Students will understand that...</p> <ul style="list-style-type: none"> <li>● Error message occur as the result of a mistake in syntax or logic.</li> <li>● Each line of code must indicate who is doing the action and what they are doing.</li> </ul>
<p><b>Areas of Focus: Proficiencies (Cumulative Progress Indicators)</b></p>	<p><b>Examples, Outcomes, Assessments</b></p>
<p>Students will:</p> <p>Be able to write valid lines of java code</p> <p>Be able to identify basic syntax errors</p> <p>Be able to compile and run their code</p>	<p>Instructional Focus (3 days):</p> <ul style="list-style-type: none"> <li>● Project 0 – Getting Started – Make the wombats pick up all the leaves in the world.</li> </ul> <p>Instructional Strategies:</p> <ul style="list-style-type: none"> <li>● Individual work</li> <li>● In-class programming time with assistance from teacher</li> </ul> <p>Technology Integration</p> <ul style="list-style-type: none"> <li>● Computer/BlueJ program</li> </ul> <p>Global Perspectives</p>

	<p>We will discuss how programming languages are the new global language. Computer languages are internationally recognized and used and learning them makes the user part of an international community. During the semester students will learn how to collaborate with others in the classroom as well as utilizing the internet as a resource for solving problems.</p>
--	---

## Unit 1: Custom Methods

Standard	
<p><b>Big Ideas:</b> <i>Course Objectives / Content Statement(s)</i></p> <ul style="list-style-type: none"> <li>● <b>Students will learn how to write their own Wombat methods utilizing the four main methods.</b></li> <li>● <b>Students will learn how to properly format a method's code.</b></li> <li>● <b>Students will learn how to identify when creating a method is necessary.</b></li> </ul>	
<p><b>Essential Questions</b> <i>What provocative questions will foster inquiry, understanding, and transfer of learning?</i></p>	<p><b>Enduring Understandings</b> <i>What will students understand about the big ideas?</i></p>
<ul style="list-style-type: none"> <li>● What is the difference between what I'm telling it to do and what I want it to do?</li> <li>● When is it required to make a method as opposed to simply writing code?</li> </ul>	<p>Students will understand that...</p> <ul style="list-style-type: none"> <li>● Methods are used to help simplify code and make it more readable.</li> <li>● They should code DRY (Don't Repeat Yourself)</li> <li>● Good code formatting is essential.</li> </ul>
<p><b>Areas of Focus: Proficiencies (Cumulative Progress Indicators)</b></p>	<p><b>Examples, Outcomes, Assessments</b></p>
<p>Students will:</p> <p>Be able to format code (brackets, variables naming, spacing, etc.)</p> <p>Be able to successfully utilize all 4 main Wombat methods</p> <p>Know good programming etiquette (camelCase, for example)</p>	<p>Instructional Focus (1 week):</p> <ul style="list-style-type: none"> <li>● Project 1 – Custom Methods – Make the wombat efficiently pick up all the leaves in the world by creating custom methods to avoid repetitious code.</li> </ul> <p>Instructional Strategies:</p> <ul style="list-style-type: none"> <li>● Individual work with help from peers</li> <li>● In-class programming time with assistance from teacher</li> <li>● At home programming time</li> </ul> <p>Technology Integration</p>

	<ul style="list-style-type: none"> <li>• Computer/BlueJ program</li> </ul> <p>Global Perspectives Students will learn to emphasize creativity in their program design. We will examine how students approach and solve similar tasks in unique ways. We will discuss what makes an “elegant” solution compared to a practical or functional solution. Students will take other student’s projects and interpret their code.</p>
--	---

## Unit 2: Conditionals

Standard	
<p><b>Big Ideas:</b> <i>Course Objectives / Content Statement(s)</i></p> <ul style="list-style-type: none"> <li>• Students will learn the 4 Wombat conditionals (<code>foundLeaf</code>, <code>hasLeaf</code>, <code>canMove</code>, <code>facingNorth</code>).</li> <li>• Students will learn how to structure an if/else block with <code>&amp;&amp;</code> and <code>  </code>.</li> </ul>	
Essential Questions <i>What provocative questions will foster inquiry, understanding, and transfer of learning?</i>	Enduring Understandings <i>What will students understand about the big ideas?</i>
<ul style="list-style-type: none"> <li>• Which parts of an if/else block are required and which are optional?</li> <li>• How does the computer know whether to run the if or the else portion of your code?</li> <li>• How can you make a multi-conditional if statement?</li> </ul>	<p>Students will understand that...</p> <ul style="list-style-type: none"> <li>• All if statements will either run the code in the if block or the else block</li> <li>• All conditionals must evaluate to true or false, no matter how complicated they are.</li> </ul>
Areas of Focus: Proficiencies (Cumulative Progress Indicators)	Examples, Outcomes, Assessments
<p>Students will:</p> <p>Be able to create as many of their own methods as they need and identify when it is appropriate to do so.</p> <p>Be able to use if statements to prevent their wombat from breaking</p>	<p>Instructional Focus (1 week):</p> <ul style="list-style-type: none"> <li>• Project 2 – Conditional Methods – Students will help a Wombat traverse the world, focusing on efficiency as part of the solution.</li> </ul> <p>Instructional Strategies:</p> <ul style="list-style-type: none"> <li>• Individual work with help from peers</li> <li>• In-class programming time with assistance from teacher</li> <li>• At home programming time</li> </ul> <p>Technology Integration</p>

- Computer/BlueJ program

#### Media Literacy Integration

We will examine how English and Java are structured differently and how the computer will interpret your directions literally. We compare an English sentence with how a computer would interpret the statement. “Pick up some milk and eggs if they have them”

## Unit 3: Generic Code

Standard	
<p><b>Big Ideas:</b> <i>Course Objectives / Content Statement(s)</i></p> <ul style="list-style-type: none"> <li>● Students will learn how to identify clean, efficiently written code.</li> <li>● Students will learn how to modify and edit their code to make it more efficient.</li> <li>● Students will be able to build methods from combinations of other methods.</li> </ul>	
Essential Questions	Enduring Understandings
<i>What provocative questions will foster inquiry, understanding, and transfer of learning?</i>	<i>What will students understand about the big ideas?</i>
<ul style="list-style-type: none"> <li>● What is the difference between a working solution and a strong/efficient solution?</li> <li>● How can I identify when a problem is fully solved?</li> </ul>	<p>Students will understand that...</p> <ul style="list-style-type: none"> <li>● Complexity is not simply based on the number of lines of code but on its run-time efficiency and how many necessary/unnecessary checks are made.</li> </ul>
Areas of Focus: Proficiencies (Cumulative Progress Indicators)	Examples, Outcomes, Assessments
<p>Students will:</p> <p>Utilize all the previously available tools at maximum effectiveness</p>	<p>Instructional Focus (1 week):</p> <ul style="list-style-type: none"> <li>● Project 3 – Generic Code – Make the Wombat pick up all the leaves in a world that is randomly generated</li> </ul> <p>Instructional Strategies:</p> <ul style="list-style-type: none"> <li>● Group work/individual work</li> <li>● In-class programming time with assistance from teacher</li> <li>● At home programming time</li> </ul> <p>Technology Integration</p> <ul style="list-style-type: none"> <li>● Computer/BlueJ program</li> </ul> <p>Global Perspectives</p> <p>Students will discuss the importance of checking and review code, as well as working with others to ensure that a project is working as intended.</p>

## Unit 4: Methods and their Return Types

Standard	
<p><b>Big Ideas:</b> <i>Course Objectives / Content Statement(s)</i></p> <ul style="list-style-type: none"> <li>● Students will learn what the ‘void’ keyword represents and when/how to use the ‘boolean’ keyword in method declarations.</li> <li>● Students will learn how to code to pre and post-conditions</li> </ul>	
Essential Questions	Enduring Understandings
<i>What provocative questions will foster inquiry, understanding, and transfer of learning?</i>	<i>What will students understand about the big ideas?</i>
<ul style="list-style-type: none"> <li>● What does it mean for a method to return something and how do I use that information?</li> <li>● What is the purpose of pre/post conditions? How are they similar and how are they different?</li> </ul>	<p>Students will understand that...</p> <ul style="list-style-type: none"> <li>● A Boolean method must return true or false.</li> <li>● That method must be present in a conditional in order to utilize the information it returns. If the information isn’t needed it should be void.</li> <li>● Pre conditions are assumed to be true, postconditions must be coded.</li> </ul>
Areas of Focus: Proficiencies (Cumulative Progress Indicators)	Examples, Outcomes, Assessments
<p>Students will:</p> <p>Be able to combine their Boolean methods with &amp;&amp; and    statements</p> <p>Be able to write methods with Boolean return types</p> <p>Be able to determine which methods require Boolean return types vs void return types.</p>	<p>Instructional Focus (1 week):</p> <ul style="list-style-type: none"> <li>● Project 4 – Methods – Students will write their own Boolean methods to help a Wombat “carpet” the world with leafs.</li> </ul> <p>Instructional Strategies:</p> <ul style="list-style-type: none"> <li>● Group work/individual work</li> <li>● In-class programming time with assistance from teacher</li> <li>● At home programming time</li> </ul> <p>Technology Integration</p> <ul style="list-style-type: none"> <li>● Computer/BlueJ program</li> </ul> <p>Media Literacy Integration</p> <p>Students will examine a basic video games’ assembly code to find cases of custom classes. Students will identify these classes and attempt to discern what it is that they do. Questions might include: “Why is this</p>

	class needed?” “How does this make the program more efficient?”
--	---

## Unit 5: Variables and Parameters

Standard	
<p><b>Big Ideas:</b> <i>Course Objectives / Content Statement(s)</i></p> <ul style="list-style-type: none"> <li>● Students will be able to create and utilize int and Boolean variables and parameters.</li> <li>● Students will learn the difference between variables and parameters.</li> <li>● Students will learn basic math (+, -, *, /, %) using ints.</li> </ul>	
Essential Questions	Enduring Understandings
<i>What provocative questions will foster inquiry, understanding, and transfer of learning?</i>	<i>What will students understand about the big ideas?</i>
<ul style="list-style-type: none"> <li>● How is java method similar and different from regular math?</li> <li>● What is the core feature difference between variables and parameters and what purpose do each of them serve?</li> <li>● How is / different from % and when should they be used?</li> </ul>	<p>Students will understand that...</p> <ul style="list-style-type: none"> <li>● Parameters require a value provided for them to function while Variables will have their value provided directly by the coder.</li> <li>● Java math has the same basic rules as regular math but only utilizing integers.</li> </ul>
Areas of Focus: Proficiencies (Cumulative Progress Indicators)	Examples, Outcomes, Assessments
<p>Students will:</p> <p>Be able to write their own Boolean return type methods, just as they had void methods prior.</p> <p>Efficiently solve individual Wombat methods (Test 1)</p>	<p>Instructional Focus (2 weeks):</p> <ul style="list-style-type: none"> <li>● Project 5 – Variables and Parameters – Students will code several individual Wombat methods utilizing variables and parameters. Each problem must be made as efficient as possible.</li> </ul> <p>Sample Assessments:</p> <ul style="list-style-type: none"> <li>● Test #1 – Students will receive 3 methods they have not yet previously seen and must code each of them appropriately.</li> </ul> <p>Instructional Strategies:</p> <ul style="list-style-type: none"> <li>● Group work/individual work</li> <li>● In-class programming time with assistance from teacher</li> <li>● At home programming time</li> </ul> <p>Technology Integration</p>

- Computer/BlueJ program

#### Media Literacy Integration

We will examine Java code from a more advanced project in order to see what parts of the project they can understand. We will attempt to demonstrate how, just like with a spoken language, you do not need to understand everything in order to figure out what is going on.

## Unit 6: Theme 1 Final Project

Standard	
<p><b>Big Ideas:</b> <i>Course Objectives / Content Statement(s)</i></p> <ul style="list-style-type: none"> <li>● <b>Students will be able to demonstrate mastery of all the previous topics in one final Wombat project.</b></li> </ul>	
Essential Questions	Enduring Understandings
<i>What provocative questions will foster inquiry, understanding, and transfer of learning?</i>	<i>What will students understand about the big ideas?</i>
<ul style="list-style-type: none"> <li>● How do I began solving a large, scalable problem?</li> <li>● How can I tell if I have an efficient solution?</li> <li>● What do I do when I don't know how to start a problem?</li> </ul>	<p>Students will understand that...</p> <ul style="list-style-type: none"> <li>● Although each of them will develop their own solution, there will be several common elements between them.</li> <li>● Code must be written so that anyone in the class could read another other student's code.</li> </ul>
Areas of Focus: Proficiencies (Cumulative Progress Indicators)	Examples, Outcomes, Assessments
<p>Students will:</p> <p>Be able to utilize one core skill from each of the previous units.</p> <p>Be able to create efficient solutions by writing and rewriting code.</p>	<p>Instructional Focus (2 weeks):</p> <ul style="list-style-type: none"> <li>● Project 6 – Maze – Students will move a Wombat from the beginning of the maze to the end, solving the problem in any fashion that they wish. Grades will be more heavily weighted based on efficiency of the solution.</li> </ul> <p>Instructional Strategies:</p> <ul style="list-style-type: none"> <li>● Group work/individual work</li> <li>● In-class programming time with assistance from teacher</li> <li>● At home programming time</li> </ul> <p>Technology Integration</p> <ul style="list-style-type: none"> <li>● Computer/BlueJ program</li> </ul> <p>Global Perspectives</p> <p>Students will investigate how programmers from different countries view problems in a different fashion and, as a result, write their programs in unique ways. We will discuss the pros and cons of attacking programs in different ways and emphasize the importance of finding unique solutions to problems.</p>

## Theme 2: Object Oriented Programming

### Unit 7: Class Creation

Standard	
<p><b>Big Ideas:</b> <i>Course Objectives / Content Statement(s)</i></p> <ul style="list-style-type: none"> <li>● Students will learn how to write a java class from scratch, including constructor, instance variables, methods and gets.</li> <li>● Students will learn the syntax of creating a class and the critical role of each component in a class.</li> <li>● Students will learn the difference between a class and an object.</li> </ul>	
Essential Questions <i>What provocative questions will foster inquiry, understanding, and transfer of learning?</i>	Enduring Understandings <i>What will students understand about the big ideas?</i>
<ul style="list-style-type: none"> <li>● What are the core components of a class and what does each component accomplish?</li> <li>● What are the computer science names for different parts of a class?</li> </ul>	<p>Students will understand that...</p> <ul style="list-style-type: none"> <li>● It is the responsibility of the programmer to devise and utilize their available tools to create a program that does what is intended.</li> <li>● They must code DRY (Don't Repeat Yourself).</li> </ul>
Areas of Focus: Proficiencies (Cumulative Progress Indicators)	Examples, Outcomes, Assessments
<p>Students will:</p> <ul style="list-style-type: none"> <li>- Be able to forward integrate all the tools they learned coding with Wombats into traditional coding.</li> <li>- Begin to use computer science terms including – instance variable, method/local variable, constructor, standard class functionality, gets/getters, etc.</li> </ul>	<p>Instructional Focus (1 week):</p> <ul style="list-style-type: none"> <li>● Project 7 – Class – Students will create a lunch account class and a variety of methods that allow students to buy lunch from the cafeteria.</li> </ul> <p>Instructional Strategies:</p> <ul style="list-style-type: none"> <li>● Individual work</li> <li>● In-class programming time with assistance from teacher</li> <li>● At home programming time</li> </ul> <p>Technology Integration</p> <ul style="list-style-type: none"> <li>● Computer/BlueJ program</li> </ul> <p>Media Literacy Integration</p> <p>Students will research languages without OOP and compare how the languages are</p>

different. We will discuss how the idea of objects is fundamental to the core of Java.

## Unit 8: Class Interaction and OOP Basics

Standard	
<p><b>Big Ideas:</b> <i>Course Objectives / Content Statement(s)</i></p> <ul style="list-style-type: none"> <li>● Students will learn how to write classes that have other classes as instance variables.</li> <li>● Students will be able to write an equals method and have an understanding of the concept of “this”</li> </ul>	
Essential Questions <i>What provocative questions will foster inquiry, understanding, and transfer of learning?</i>	Enduring Understandings <i>What will students understand about the big ideas?</i>
<ul style="list-style-type: none"> <li>● What is “this” in Java and what is its purpose? What does it represent?</li> <li>● How is this project similar to our Wombat projects that had numerous Wombats interacting with each other?</li> <li>● How does the computer know which lunch object is being acted on for each line of code?</li> <li>● What are the major inefficiencies in this type of programming?</li> </ul>	<p>Students will understand that...</p> <ul style="list-style-type: none"> <li>● Classes may contain other classes as variables.</li> <li>● Code should utilize their object’s methods to create efficient code.</li> </ul>
Areas of Focus: Proficiencies (Cumulative Progress Indicators)	Examples, Outcomes, Assessments
<p>Students will:</p> <ul style="list-style-type: none"> <li>- Be able to quickly write simple classes from scratch without errors.</li> <li>- They will be able to identify which instance variables are required to solve the indicated task.</li> <li>- Be able to speculate on how this code could be improves using tools learned in the “Introduction to Computer Programming” class.</li> </ul>	<p>Instructional Focus (2 weeks):</p> <ul style="list-style-type: none"> <li>● Project 8 – Cafeteria – After fixing project 7, they will integrate this project into a new class that has 4 lunch accounts as it’s instance variables.</li> </ul> <p>Sample Assessments:</p> <ul style="list-style-type: none"> <li>● Test #2 – Students will code an entire class from scratch along with identifying required instance variables and a number of small methods.</li> </ul> <p>Instructional Strategies:</p> <ul style="list-style-type: none"> <li>● Individual work</li> </ul>

- In-class programming time with assistance from teacher
- At home programming time

#### Technology Integration

- Computer/BlueJ program

#### Media Literacy Integration

Students will examine and discuss the great Zune crash. Around the entire globe on one day, every single Zune simultaneously crashed, costing Microsoft millions of dollars to fix. The problem ended up being a single line of code.

## Unit 9: Introduction to Arrays

Standard	
<p><b>Big Ideas:</b> <i>Course Objectives / Content Statement(s)</i></p> <ul style="list-style-type: none"> <li>● Students will learn how to use for loops to iterate through an array.</li> <li>● Students will learn how to create and manipulate arrays and their various attributes.</li> </ul>	
<p><b>Essential Questions</b></p> <p><i>What provocative questions will foster inquiry, understanding, and transfer of learning?</i></p>	<p><b>Enduring Understandings</b></p> <p><i>What will students understand about the big ideas?</i></p>
<ul style="list-style-type: none"> <li>● How does an array simplify our previously written code?</li> <li>● How are arrays and for loops linked? How are they not?</li> </ul>	<p>Students will understand that...</p> <ul style="list-style-type: none"> <li>● Array's lengths are static. While the elements in the array may be changed, the number of elements may not.</li> <li>● A for loop assists in coding with arrays but is not required.</li> </ul>
<p><b>Areas of Focus: Proficiencies (Cumulative Progress Indicators)</b></p>	<p><b>Examples, Outcomes, Assessments</b></p>
<p>Students will:</p> <ul style="list-style-type: none"> <li>- Be able to identify inefficient code and simplify it.</li> <li>- Be able to identify which code requires the use of a for loop and which does not</li> <li>- Be able to create basic array algorithms.</li> </ul>	<p>Instructional Focus (2 weeks):</p> <ul style="list-style-type: none"> <li>● Project 9 – ArrayWork – Students will solve a series of small methods providing them the opportunity to work with arrays.</li> </ul> <p>Instructional Strategies:</p> <ul style="list-style-type: none"> <li>● Individual work</li> <li>● In-class programming time with assistance from teacher</li> <li>● At home programming time</li> </ul> <p>Technology Integration</p> <ul style="list-style-type: none"> <li>● Computer/BlueJ program</li> </ul>

## Unit 10: Introduction to Strings

Standard	
<p><b>Big Ideas:</b> <i>Course Objectives / Content Statement(s)</i></p> <ul style="list-style-type: none"> <li>● Students will be able to solve problems utilizing Strings (substring, equals, length methods)</li> <li>● Students will learn the relationship between arrays and Strings</li> <li>● Students will learn that String are immutable.</li> </ul>	
Essential Questions	Enduring Understandings
<i>What provocative questions will foster inquiry, understanding, and transfer of learning?</i>	<i>What will students understand about the big ideas?</i>
<ul style="list-style-type: none"> <li>● How are String stored in the computer’s memories? Why are String not “built-in” to Java?</li> <li>● How are String and Arrays similar? How are they different?</li> </ul>	<p>Students will understand that...</p> <ul style="list-style-type: none"> <li>● Strings cannot be changed, but they may be redefined.</li> <li>● All classes have an equals method to utilize.</li> </ul>
Areas of Focus: Proficiencies (Cumulative Progress Indicators)	Examples, Outcomes, Assessments
<p>Students will:</p> <ul style="list-style-type: none"> <li>- Be able to draw strong connections between their array solutions and their string solutions.</li> <li>- Solve problems that integrate both strings and arrays without direct instruction.</li> </ul>	<p>Instructional Focus (2 weeks):</p> <ul style="list-style-type: none"> <li>● Project 10 – StringWork – Students will solve a variety of small, increasingly difficult String methods.</li> </ul> <p>Sample Assessments:</p> <ul style="list-style-type: none"> <li>● <b>Test #3:</b> Students will solve 4 new array/string methods from scratch including one method that integrates the two concepts.</li> </ul> <p>Instructional Strategies:</p> <ul style="list-style-type: none"> <li>● Individual work</li> <li>● In-class programming time with assistance from teacher</li> <li>● At home programming time</li> </ul> <p>Technology Integration</p> <ul style="list-style-type: none"> <li>● Computer/BlueJ program</li> </ul> <p>Media Literacy Integration</p> <p>Students will explore the various classes that are available in Java and their uses. They will not need to code with these classes but simply are to become aware of the possibilities of working in Java.</p>

## Unit 11: Multiple Class Interactions and OOP Concepts

Standard	
<p><b>Big Ideas:</b> <i>Course Objectives / Content Statement(s)</i></p> <ul style="list-style-type: none"> <li>● Students will be able to integrate the concepts of the previous three units together to create a single project that utilizes arrays of objects, toString and multiple classes interacting with each other.</li> <li>● Students will be able to code setters and toString in addition to gets and equals.</li> </ul>	
Essential Questions <i>What provocative questions will foster inquiry, understanding, and transfer of learning?</i>	Enduring Understandings <i>What will students understand about the big ideas?</i>
<ul style="list-style-type: none"> <li>● How do these previously learned elements create an effective, efficient method of problem solving?</li> <li>● How could previous projects be made more efficient using these methods?</li> </ul>	<p>Students will understand that...</p> <ul style="list-style-type: none"> <li>● By default, S.O.P will display an objects memory location and that writing a toString will replace that information.</li> <li>● All classes should have the minimum number of instance variables required to solve the problem.</li> </ul>
Areas of Focus: Proficiencies (Cumulative Progress Indicators)	Examples, Outcomes, Assessments
<p>Students will:</p> <ul style="list-style-type: none"> <li>- Be able to code an entire class from scratch writing gets, sets, toString and equals.</li> <li>- Be able to create an array of objects and iterate through them.</li> </ul>	<p>Instructional Focus (2 weeks):</p> <ul style="list-style-type: none"> <li>● Project 11 – Class Interaction – Students will create a class to hold information and then create a larger containing class with an array of objects. They will code gets/sets/toString/equals as well.</li> </ul> <p>Sample Assessments:</p> <ul style="list-style-type: none"> <li>● Test #4 – Students will demonstrate mastery of these core concepts by coding simple class interactions together.</li> </ul> <p>Instructional Strategies:</p> <ul style="list-style-type: none"> <li>● Individual work</li> <li>● In-class programming time with assistance from teacher</li> <li>● At home programming time</li> </ul> <p>Technology Integration</p> <ul style="list-style-type: none"> <li>● Computer/BlueJ program</li> </ul>

## Unit 12: Integration of Arrays, Strings and OOP Concepts

Standard	
<p><b>Big Ideas:</b> <i>Course Objectives / Content Statement(s)</i></p> <ul style="list-style-type: none"> <li>● Students will be able to fully integrate all topics learned this year into a final project.</li> <li>● Students will be able to create multiple interacting classes from scratch all following good programming standards.</li> </ul>	
Essential Questions	Enduring Understandings
<i>What provocative questions will foster inquiry, understanding, and transfer of learning?</i>	<i>What will students understand about the big ideas?</i>
<ul style="list-style-type: none"> <li>● How can I organize my work so that I do not become confused?</li> <li>● Which tasks should I tackle first? How can I tell the ideal order?</li> </ul>	<p>Students will understand that...</p> <ul style="list-style-type: none"> <li>● A tight, consistently coded project develop to specifications is a goal that should always be strived for.</li> </ul>
Areas of Focus: Proficiencies (Cumulative Progress Indicators)	Examples, Outcomes, Assessments
<p>Students will:</p> <ul style="list-style-type: none"> <li>- Demonstrate mastery of all previously taught topics in this final project.</li> </ul>	<p>Instructional Focus (2 weeks):</p> <ul style="list-style-type: none"> <li>● Project 12 – Multiple-class Interaction – Students will modify their previous project to integrate multiple classes and multiple arrays of objects.</li> </ul> <p>Instructional Strategies:</p> <ul style="list-style-type: none"> <li>● Individual work</li> <li>● In-class programming time with assistance from teacher</li> <li>● At home programming time</li> </ul> <p>Technology Integration</p> <ul style="list-style-type: none"> <li>● Computer/BlueJ program</li> </ul> <p>Media Literacy Integration</p> <p>Students will examine, and fix, the code of students of a similar ability level to themselves. They will bring this code up to standards.</p>