



PISCATAWAY TOWNSHIP SCHOOLS

Dr. Frank Ranelli

Superintendent of Schools

Dr. William Baskerville

Assistant Superintendent

Introduction to Computer Science

Content Area: Mathematics

Grade Span: 9-12

Melissa Eytchison

Akshay Patel

Revised by: Brinder Soin

Title **Teacher**

Approval Date: August 2021

Members of the Board of Education

Shelia Hobson, President

Kimberly Lane, Vice President

Shantell Cherry

Jeffrey Fields

Ralph Johnson

Calvin Laughlin

Nitang Patel

Zoe Scotto

Brenda Smith

Piscataway Township Schools

1515 Stelton Road

Piscataway, NJ 08854-1332

732 572-2289, ext. 2561

Fax 732 572-1540

www.piscatawayschools.org

COURSE OVERVIEW

Description

This course is designed to introduce students to logical thinking, problem solving and fundamentals of computer science. This course is taught in two semesters:

Semester I: During the first part of the course students learn fundamentals of problem solving by creating games, animation and applications using Alice. Alice is an environment that combines the ideas of film making and computer science to provide students with a unique experience.

Semester II: During the part of the course, students are introduced to the JAVA programming language. Students apply the logic they have developed during the first part of the year to write formal code, develop solutions to various problems, and to grasp understanding of the syntax.

There are no prerequisites for this course.

Goals

This course is designed to introduce students to computers, computer programming, and the field of computer science. Upon completion of the course, students will understand the (1) architecture of a computer, (2) how to use computers and computer programming to solve real-world problems, and (3) how to program in JAVA programming language. This course serves as the foundation for : AP Computer Science A, and AP Computer Science Principle courses.

Scope and Sequence

Unit	Topic	Length
Alice Chapter 1	Getting Started with Alice	7 Days
Alice Chapter 2	Methods	10 Days
Alice Chapter 3	Functions, Variables and Expressions	11 Days
Alice Chapter 4	Flow Control	11 Days
Java Unit 1	Binary/Decimal Conversion Getting Started with Java (Writing Classes)	8 Days
Java Unit 2	Scanner Class, Math Class, and Data Types	14 Days
Java Unit 3	String Class	14 Days
Java Unit 4	Flow Control	14 Days

Resources

Core Text: Alice 3 in Action with Java-Joel Adams

Alice Chapter 1:

Summary and Rationale

In this chapter, students will build their first computer program using Alice, a free software tool for creating virtual worlds.

Developing programs to solve problems is a complex process that is both an art and a science. It is an art in that it requires a good deal of imagination, creativity, and ingenuity. But it is also a science in that it uses certain techniques and methodologies. In this chapter, students will work through the thought process that goes into creating computer software.

This chapter marks the beginning of the problem-solving thinking that students develop in this course. Focus in this chapter is on developing understanding for sequencing by writing algorithms and getting familiar with the Alice IDE in a fun manner.

Recommended Pacing

7 days (1 day = 80 minutes)

State Standards

Standard--Algorithms & Programming

CPI #	Cumulative Progress Indicator (CPI)
8.1.1 2.AP. 1	Design algorithms to solve computational problems using a combination of original and existing algorithms
8.1.1 2.AP. 3	Select and combine control structures for a specific application based upon performance and readability, and identify trade-offs to justify the choice.
8.1.1 2.AP. 4	Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue

Standard --Engineering Design

CPI #	Cumulative Progress Indicator (CPI)
8.2.2. ED.1	Communicate the function of a product or device
8.2.2. ED.2	Collaborate to solve a simple problem, or to illustrate how to build a product using the design process
8.2.2. ED.3	Select and use appropriate tools and materials to build a product using the design process.

Instructional Focus

Unit Enduring Understandings

- Incorporating multiple perspectives through collaboration improves programming design.
- Developers create and innovate using an iterative design process that is user-focused, that incorporates implementation/feedback cycles, and that leaves ample room for experimentation and risk-taking

- Programs can be used to process data, which allows users to discover information and create new knowledge.
- The way statements are sequenced and combined in a program determines the computed result. Programs incorporate iteration and selection constructs to represent repetition and make decisions to handle varied input values.
- To find specific solutions to generalizable problems, programmers represent and organize data in multiple ways
- Programmers break down problems into smaller and more manageable pieces. By creating procedures and leveraging parameters, programmers generalize processes that can be reused. Procedures allow programmers to draw upon existing code that has already been tested, allowing them to write programs more quickly and with more confidence.

Unit Essential Questions

- What is an algorithm?
- What is a program?
- What is an object oriented language?
- What is a statement?
- What is software design?
- What is a class?
- What determines the flow of a program?
- What is a method?
- What is Software Engineering?
- How do you animate objects in Alice?
- Why do you use doInOrder and doTogether?
- What is quadview?
- What is a user story?
- What is an IDE?

Objectives

Students will know:

- How Software Engineering works
- How to design, build, and execute Alice Program
- How Alice is broken down in classes and relate it to Java.

Students will be able to:

- Compile and Run a simple Alice Programs
- Use Alice controls like do together and do in order
- Change an Object's properties from within a program
- Uses different Alice views to position objects near one another

Resources

Core Text: Alice 3 in Action with Java-Joel Adams

Alice Chapter 2:

Summary and Rationale

There are two different reasons for building your own methods. One reason is to divide your story into smaller pieces to keep it more manageable and organized. A second reason is to provide an object with a behavior it needs but does not already have. In this chapter, you will examine both approaches. Note that the motivations, thought processes, and circumstances are quite different for these two approaches. This chapter focuses on developing understanding for the need of methods and functions in procedures. Using the divide and conquer approach students learn to divide a longer task into reusable and manageable pieces.

Recommended Pacing

10 days (1 day = 80 minutes)

State Standards

Standard--Algorithms & Programming

CPI #	Cumulative Progress Indicator (CPI)
8.1.1 2.AP. 1	Design algorithms to solve computational problems using a combination of original and existing algorithms
8.1.1 2.AP. 3	Select and combine control structures for a specific application based upon performance and readability, and identify trade-offs to justify the choice.
8.1.1 2.AP. 4	Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue

Standard --Engineering Design

CPI #	Cumulative Progress Indicator (CPI)
8.2.2. ED.1	Communicate the function of a product or device
8.2.2. ED.2	Collaborate to solve a simple problem, or to illustrate how to build a product using the design process
8.2.2. ED.3	Select and use appropriate tools and materials to build a product using the design process.

Instructional Focus

Unit Enduring Understandings

- Incorporating multiple perspectives through collaboration improves programming design.
- Developers create and innovate using an iterative design process that is user-focused, that incorporates implementation/feedback cycles, and that leaves ample room for experimentation and risk-taking
- Programs can be used to process data, which allows users to discover information and create new knowledge.
- The way statements are sequenced and combined in a program determines the computed result. Programs incorporate iteration and selection constructs to represent repetition and make decisions to handle varied input values.
- To find specific solutions to generalizable problems, programmers represent and organize data in multiple ways

- Programmers break down problems into smaller and more manageable pieces. By creating procedures and leveraging parameters, programmers generalize processes that can be reused. Procedures allow programmers to draw upon existing code that has already been tested, allowing them to write programs more quickly and with more confidence.

Unit Essential Questions

- What is an algorithm?
- What is abstraction?
- What are Methods?
- How do you create a Method in Alice?
- When are World Methods used?
- When are Object Methods used?
- Why are names important?
- What are Scenes used for?
- When are Shots necessary?
- What is a marker and how do we use it?
- What is an object's point of view?
- How do you move the camera?

Objectives

Students will know:

- How to break a story into scenes and shots and relate it to coding.
- How to build class methods
- How to reposition Camera for different views

Students will be able to:

- Build scene methods to help organize a story into scenes and shots
- Build class methods to elicit desirable behaviors from objects
- Use markers to reposition the camera for different shots within a scene
- Understand how an object's position, orientation, and point of view are determined

Resources

Core Text: Alice 3 in Action with Java-Joel Adams

Alice Chapter 3:

Summary and Rationale

Alice provides three kinds of variables, all of which will be examined in this chapter. The first kind is the local variable, which lets us compute and store a value within a method for later use. Local variables are created using the variable. . . control at the bottom-right of the editing area.

The second kind of variable is the parameter, which lets us write methods that are more broadly useful. Parameters are created using the Add Parameter. . . button, which is located inside the parentheses of Alice’s programmer-defined methods.

The third and final kind of variable is the instance variable or property, which lets us store a characteristic of an object. Properties are declared in an object’s class, so property variables can be created using the Add Property. . . button at the bottom of a class page, accessed via the class navigator.

In this chapter, students will learn how to create and use all three kinds of variables.

Advancing the problem-solving process, this chapter introduces students to variables, which are used to remember information in a program. Variables are further used to make functions and methods more general in the sense that same function returns different output based upon the input.

Recommended Pacing

11 days (1 day = 80 minutes)

State Standards

Standard--Algorithms & Programming

CPI #	Cumulative Progress Indicator (CPI)
8.1.1 2.AP. 1	Design algorithms to solve computational problems using a combination of original and existing algorithms
8.1.1 2.AP. 3	Select and combine control structures for a specific application based upon performance and readability, and identify trade-offs to justify the choice.
8.1.1 2.AP. 4	Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue
8.1.1 2.AP. 7	Collaboratively design and develop programs and artifacts for broad audiences by incorporating feedback from users

Standard --Engineering Design

CPI #	Cumulative Progress Indicator (CPI)
8.2.2. ED.1	Communicate the function of a product or device
8.2.2. ED.2	Collaborate to solve a simple problem, or to illustrate how to build a product using the design process

8.2.2. ED.3	Select and use appropriate tools and materials to build a product using the design process.
Instructional Focus	
Unit Enduring Understandings	
<ul style="list-style-type: none"> ● Incorporating multiple perspectives through collaboration improves programming design. ● Developers create and innovate using an iterative design process that is user-focused, that incorporates implementation/feedback cycles, and that leaves ample room for experimentation and risk-taking ● Programs can be used to process data, which allows users to discover information and create new knowledge. ● The way statements are sequenced and combined in a program determines the computed result. Programs incorporate iteration and selection constructs to represent repetition and make decisions to handle varied input values. ● To find specific solutions to generalizable problems, programmers represent and organize data in multiple ways ● Programmers break down problems into smaller and more manageable pieces. By creating procedures and leveraging parameters, programmers generalize processes that can be reused. Procedures allow programmers to draw upon existing code that has already been tested, allowing them to write programs more quickly and with more confidence. 	
Unit Essential Questions	
<ul style="list-style-type: none"> • What is an algorithm? • What is abstraction? • What are Functions? • How do you create a function in Alice? • When are World Functions used? • When are Object Functions used? • What is the difference between a function and a method? • How are functions and methods similar? • When do you use variables? • When do you use parameters? • Why are names important? • What are Alice data types? • How do you build an expression in Alice? • What is an object's vehicle? • How do you get information from the user in Alice? • How do you concatenate in Alice? 	
Objectives	
<p>Students will know:</p> <ul style="list-style-type: none"> ● How to create and use variables and parameters ● How to create and use functions <p>Students will be able to:</p> <ul style="list-style-type: none"> ● Use variables to store values that can be used later in a method ● Use a variable to store the value of an arithmetic expression ● Use a variable to store the value produced by a function ● Use parameters to write methods that are more broadly useful ● Define and access property variables ● Create functions-messages that return a value to their sender 	
Resources	

Alice Chapter 4:

Summary and Rationale

In this chapter, students will examine several of Alice’s flow control statements, including:

- The **if** statement, which directs the flow through one group of statements and away from another group of statements
- The **for** statement, which directs the flow through a group of statements a fixed number of times
- The **while** statement, which directs the flow through a group of statements an arbitrary number of times

This chapter marks the most important step in the problem-solving process. Here students develop the art of writing flow charts, understand how to control the follow of process using conditional and repetition structures.

Recommended Pacing

11 days (1 day = 80 minutes)

State Standards

Standard--Algorithms & Programming

CPI #	Cumulative Progress Indicator (CPI)
8.1.1 2.AP. 1	Design algorithms to solve computational problems using a combination of original and existing algorithms
8.1.1 2.AP. 3	Select and combine control structures for a specific application based upon performance and readability, and identify trade-offs to justify the choice.
8.1.1 2.AP. 4	Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue
8.1.1 2.AP. 7	Collaboratively design and develop programs and artifacts for broad audiences by incorporating feedback from users

Standard --Engineering Design

CPI #	Cumulative Progress Indicator (CPI)
8.2.2. ED.1	Communicate the function of a product or device
8.2.2. ED.2	Collaborate to solve a simple problem, or to illustrate how to build a product using the design process
8.2.2. ED.3	Select and use appropriate tools and materials to build a product using the design process.

Instructional Focus

Unit Enduring Understandings

- Incorporating multiple perspectives through collaboration improves programming design.
- Developers create and innovate using an iterative design process that is user-focused, that incorporates implementation/feedback cycles, and that leaves ample room for experimentation and risk-taking
- Programs can be used to process data, which allows users to discover information and create new knowledge.
- The way statements are sequenced and combined in a program determines the computed result. Programs incorporate iteration and selection constructs to represent repetition and make decisions to handle varied input values.
- To find specific solutions to generalizable problems, programmers represent and organize data in multiple ways
- Programmers break down problems into smaller and more manageable pieces. By creating procedures and leveraging parameters, programmers generalize processes that can be reused. Procedures allow programmers to draw upon existing code that has already been tested, allowing them to write programs more quickly and with more confidence.

Unit Essential Questions

- What is an algorithm?
- What is abstraction?
- How do you control the flow of a program?
- What is selective execution?
- How does iteration work?
- When should you use a For Loop?
- When should you use a While Loop?
- What is nesting?
- What are Boolean expressions/variables?
- What are relational operators?
- What are boolean operators?

Objectives

Students will know:

- How to use different Boolean type flow control operators
- How to use conditional statements like “If-Then”

Students will be able to:

- Use the Boolean type and its basic operation
- Use the if statement to perform some statements while skipping others
- Use the for and while statements to perform (other) statements more than once
- Use Boolean variables and functions to control if and while statements

Resources

Core Text: Alice 3 in Action with Java-Joel Adams

Java Unit 1:

Summary and Rationale	
<p>In this chapter, students will begin to study programming in Java. Accordingly, this chapter will be used to examine designing, writing, running, and testing Java programs.</p> <p>In addition to moving from Alice to Java, students will also develop understanding of how various programming languages interact with the computer. In this chapter students are also taught various number systems: Binary, Octal Decimal and HexaDecimal.</p>	
Recommended Pacing	
8 days (1 day = 80 minutes)	
State Standards	
Standard--Algorithms & Programming	
CPI #	Cumulative Progress Indicator (CPI)
8.1.1 2.AP. 1	Design algorithms to solve computational problems using a combination of original and existing algorithms
8.1.1 2.AP. 3	Select and combine control structures for a specific application based upon performance and readability, and identify trade-offs to justify the choice.
8.1.1 2.AP. 4	Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue
Standard --Engineering Design	
CPI #	Cumulative Progress Indicator (CPI)
8.2.2. ED.1	Communicate the function of a product or device
8.2.2. ED.2	Collaborate to solve a simple problem, or to illustrate how to build a product using the design process
8.2.2. ED.3	Select and use appropriate tools and materials to build a product using the design process.
Instructional Focus	
Unit Enduring Understandings	

- The way a computer represents data internally is different from the way the data are interpreted and displayed for the user. Programs are used to translate data into a representation more easily understood by people.
- Incorporating multiple perspectives through collaboration improves programming design.
- Developers create and innovate using an iterative design process that is user-focused, that incorporates implementation/feedback cycles, and that leaves ample room for experimentation and risk-taking
- Programs can be used to process data, which allows users to discover information and create new knowledge.
- The way statements are sequenced and combined in a program determines the computed result. Programs incorporate iteration and selection constructs to represent repetition and make decisions to handle varied input values.
- To find specific solutions to generalizable problems, programmers represent and organize data in multiple ways
- Programmers break down problems into smaller and more manageable pieces. By creating procedures and leveraging parameters, programmers generalize processes that can be reused. Procedures allow programmers to draw upon existing code that has already been tested, allowing them to write programs more quickly and with more confidence.

Unit Essential Questions

- What is an algorithm?
- What is an Eclipse/Java program?
- What is an object oriented language?
- What is a statement?
- How do you convert between decimal and binary numbers?
- Why do computers use binary?
- What is a class in Eclipse/Java?
- How are text-based programming languages different from block-based programming languages?
- What is a compiler?
- What is an IDE?
- What are syntax errors?
- What are logic errors?
- How do you output information?

Objectives

Students will know:

- Understand how binary numbers are used in Computer Programming
- What an IDE is
- Transition from Alice to Java
- Different classes in Java

Students will be able to:

- Convert from decimal to binary and vice versa
- Compile and execute first program in Java
- Learn the basics of the Eclipse integrated development environment (IDE)
- Identify and fix errors in Java

Resources

Core Text: Alice 3 in Action with Java-Joel Adams

Java Unit 2:

Summary and Rationale	
<p>In this chapter, students will examine the concept of type in greater depth, to see why it is so important in programming. Along the way, they will learn how to declare variables and constants, how to construct expressions that set their values, learn more about the placeholders <code>printf()</code> uses, and learn how to use some of the functions that Java provides.</p> <p>In addition students also learn how to receive different types of input from the user, creating and using objects for the pre existing classes, and writing mathematical expressions.</p>	
Recommended Pacing	
14 days (1 day = 80 minutes)	
State Standards	
Standard--Algorithms & Programming	
CPI #	Cumulative Progress Indicator (CPI)
8.1.1 2.AP. 1	Design algorithms to solve computational problems using a combination of original and existing algorithms
8.1.1 2.AP. 3	Select and combine control structures for a specific application based upon performance and readability, and identify trade-offs to justify the choice.
8.1.1 2.AP. 4	Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue
8.1.1 2.AP. 7	Collaboratively design and develop programs and artifacts for broad audiences by incorporating feedback from users
Standard --Engineering Design	
CPI #	Cumulative Progress Indicator (CPI)

8.2.2. ED.1	Communicate the function of a product or device
8.2.2. ED.2	Collaborate to solve a simple problem, or to illustrate how to build a product using the design process
8.2.2. ED.3	Select and use appropriate tools and materials to build a product using the design process.

Instructional Focus

Unit Enduring Understandings

- Incorporating multiple perspectives through collaboration improves programming design.
- Developers create and innovate using an iterative design process that is user-focused, that incorporates implementation/feedback cycles, and that leaves ample room for experimentation and risk-taking
- Programs can be used to process data, which allows users to discover information and create new knowledge.
- The way statements are sequenced and combined in a program determines the computed result. Programs incorporate iteration and selection constructs to represent repetition and make decisions to handle varied input values.
- To find specific solutions to generalizable problems, programmers represent and organize data in multiple ways
- Programmers break down problems into smaller and more manageable pieces. By creating procedures and leveraging parameters, programmers generalize processes that can be reused. Procedures allow programmers to draw upon existing code that has already been tested, allowing them to write programs more quickly and with more confidence.

Unit Essential Questions

- What is Scanner in Eclipse/Java?
- What are import statements?
- How do you create an object in Eclipse?
- Why is naming important?
- How do you define variables in Java?
- How do you use Math in Java?
- How do decimals work in Java?
- Does data type matter?
- What are primitive data types?
- How is memory associated with data types?
- What is the Java API?

Objectives

Students will know:

- How to get information from the user
- How to define variables and constants
- How to write complex Java program
- What Java API is
- The different data types

Students will be able to:

- Describe and use Java's Scanner and PrintStream Classes
- Build Complex Java expressions
- Define Variables in Java
- Use Math in writing Java programs
- Create an Object in Eclipse

Resources

Java Unit 3:

Summary and Rationale	
<p>During this chapter students should begin to feel comfortable with the declaration of variables, and discuss what happens at each point in declaring an object, creating an instance of that object, assigning vs. changing the state of the object and how that differs from what happens with primitive data. Students will begin to do mathematical calculations and should come to understand both the mathematical operators and their precedence.</p>	
Recommended Pacing	
14 days (1 day = 80 minutes)	
State Standards	
Standard--Algorithms & Programming	
CPI #	Cumulative Progress Indicator (CPI)
8.1.1 2.AP. 1	Design algorithms to solve computational problems using a combination of original and existing algorithms
8.1.1 2.AP. 3	Select and combine control structures for a specific application based upon performance and readability, and identify trade-offs to justify the choice.
8.1.1 2.AP. 4	Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue
8.1.1 2.AP. 7	Collaboratively design and develop programs and artifacts for broad audiences by incorporating feedback from users
Standard --Engineering Design	
CPI #	Cumulative Progress Indicator (CPI)

8.2.2. ED.1	Communicate the function of a product or device
8.2.2. ED.2	Collaborate to solve a simple problem, or to illustrate how to build a product using the design process
8.2.2. ED.3	Select and use appropriate tools and materials to build a product using the design process.
Instructional Focus	
Unit Enduring Understandings	
<ul style="list-style-type: none"> ● Incorporating multiple perspectives through collaboration improves programming design. ● Developers create and innovate using an iterative design process that is user-focused, that incorporates implementation/feedback cycles, and that leaves ample room for experimentation and risk-taking ● Programs can be used to process data, which allows users to discover information and create new knowledge. ● The way statements are sequenced and combined in a program determines the computed result. Programs incorporate iteration and selection constructs to represent repetition and make decisions to handle varied input values. ● To find specific solutions to generalizable problems, programmers represent and organize data in multiple ways ● Programmers break down problems into smaller and more manageable pieces. By creating procedures and leveraging parameters, programmers generalize processes that can be reused. Procedures allow programmers to draw upon existing code that has already been tested, allowing them to write programs more quickly and with more confidence. 	
Unit Essential Questions	
<ul style="list-style-type: none"> • What is String? • How is a String data type different from a primitive data type? • How do String methods work? • How do you set up a String object? • How is lexicography related to String methods? 	
Objectives	
<p>Students will know:</p> <ul style="list-style-type: none"> • What string are • Different string methods • Difference between literal string vs String object <p>Students will be able to:</p> <ul style="list-style-type: none"> ● Write a program using string class ● Define a variable of String type ● Use string methods to write a program and run it 	
Resources	
Core Text: Alice 3 in Action with Java-Joel Adams	

Java Unit 4:

Summary and Rationale	
<p>In the last chapter, we saw how to define Java methods that run sequentially from beginning to end. In this chapter, students will be introduced to Java’s if, switch, while, for, and do statements, which will be used to write methods that provide more complex behaviors.</p> <p>This chapter deals with program structure and control. As a part of this chapter students should become familiar with both decision and looping structures and the process of decomposing a problem to decide where these structures are necessary. The use of flow charts and syntax diagrams are used throughout the text and can be integrated into this chapter as a visual representation of the processes for students. Also students need to be able to recognize situations where loops and decision structures are necessary.</p>	
Recommended Pacing	
14 days (1 day = 80 minutes)	
State Standards	
Standard--Algorithms & Programming	
CPI #	Cumulative Progress Indicator (CPI)
8.1.1 2.AP. 1	Design algorithms to solve computational problems using a combination of original and existing algorithms
8.1.1 2.AP. 3	Select and combine control structures for a specific application based upon performance and readability, and identify trade-offs to justify the choice.

8.1.1 2.AP. 4	Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue
8.1.1 2.AP. 7	Collaboratively design and develop programs and artifacts for broad audiences by incorporating feedback from users
Standard --Engineering Design	
CPI #	Cumulative Progress Indicator (CPI)
8.2.2. ED.1	Communicate the function of a product or device
8.2.2. ED.2	Collaborate to solve a simple problem, or to illustrate how to build a product using the design process
8.2.2. ED.3	Select and use appropriate tools and materials to build a product using the design process.
Instructional Focus	
Unit Enduring Understandings	
<ul style="list-style-type: none"> ● Incorporating multiple perspectives through collaboration improves programming design. ● Developers create and innovate using an iterative design process that is user-focused, that incorporates implementation/feedback cycles, and that leaves ample room for experimentation and risk-taking ● Programs can be used to process data, which allows users to discover information and create new knowledge. ● The way statements are sequenced and combined in a program determines the computed result. Programs incorporate iteration and selection constructs to represent repetition and make decisions to handle varied input values. ● To find specific solutions to generalizable problems, programmers represent and organize data in multiple ways ● Programmers break down problems into smaller and more manageable pieces. By creating procedures and leveraging parameters, programmers generalize processes that can be reused. Procedures allow programmers to draw upon existing code that has already been tested, allowing them to write programs more quickly and with more confidence. 	
Unit Essential Questions	
<ul style="list-style-type: none"> • What is an algorithm? • What is abstraction? • How do you control the flow of a program? • What is selective execution in Java? • How does iteration work in Java? • When should you use a For Loop? • When should you use a While Loop? • What is nesting? • What are Boolean expressions/variables? • What are relational operators? • What are boolean operators? 	
Objectives	
Students will know:	
<ul style="list-style-type: none"> • How to use conditional statements in Java to write a program • Definite variables of Boolean types • Understand different boolean operators 	

Students will be able to:

- Write a code using Boolean statement
- Write a code involving finite and infinite loops
- Use Boolean operators in the program
- Use abstraction to write code

Resources

Core Text: Alice 3 in Action with Java-Joel Adams