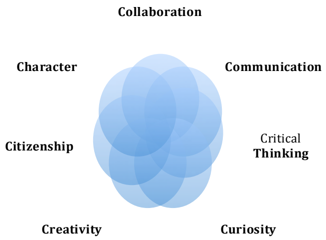
Region 14 Schools
Inspiring Excellence

Content Area: Mathematics	Course: Computer Programming	Grade Level: 11-12
	R14 The Seven Cs of Learning  Collaboration Character Communication Citizenship Critical Thinking Creativity Curiosity	

Unit Titles	Length of Unit
● Introduction to Programming with Java	1 week
● Working with Types of Data	1-2 weeks
● Branches and Loops	3-4 weeks
● File Input and Output	1-2 weeks
● Object Oriented Programming	4-5 weeks
● More Java Features	2-3 weeks
● Graphics and Arrays	3-4 weeks
● Graphical User Interface Programming	3 weeks

Strands	Course Level Expectations
***	CT State Standards for Computer Programming are not yet available, however, this curriculum has been modeled following select topics and course expectations from the AP Computer Science course.

Unit Title	Introduction to Programming with Java	Length of Unit	1 weeks

Inquiry Questions (Engaging & Debatable)	● What is computer programming and why is it important? ● How is Java better or worse than other programming languages? ● What are the advantages and disadvantages of object-oriented programming?
Standards	*** See Above
Unit Strands & Concepts	● Hello World program, Java Bytecodes, Java compiler and Java virtual machine ● Using Notepad to create a source program ● Running a Java program, Small example programs ● Names for source files and class files ● Syntax errors and Bugs ● The Edit, Compile, and Run cycle ● Matching braces and Neat indenting ● How to copy, paste, compile, and run programs ● Starting Notepad and the command prompt ● Copying text to the clipboard and pasting text from the clipboard to Notepad ● Saving the text as a source file
Key Vocabulary	Source Program, Bytecode, Javac, Emulator, Portability, Applet, Case Sensitive, Class, Identifier, Reserved Word, Method, Main Method, String Literal, Syntax Error, Bug, Comment, Copy, Paste, Save, Run

Unit Title	Introduction to Programming with Java	**Length of Unit**	1 weeks

Critical Content: **My students will Know…**	**Key Skills:** **My students will be able to (Do)…**
• Java converts source programs into bytecodes. • Javac is the name of the compiler that makes this conversion. • We use emulator software, in the absence of a Java processor chip, in order to compile and run a java program on any computer. • An applet is a Java Bytecode program that runs on a Web browser. • A bug is different than an error, because a bug still allows a program to compile but it doesn't act as intended when it runs.	• Write, compile, and run a simple program entitled, "Hello". • Choose appropriate identifiers for a class. • Begin to search for and correct syntax errors in simple programs. • Include comments in simple programs that will allow them to indicate the various aspects and intent of the programming code. • Copy and Paste in source code and Save, Compile, and Run programs.

Assessments:	• Program writing activities including, but not limited to, math related algorithms and challenges from projecteuler.net • Formative Assessment to be created and linked to an implementation guide
Teacher Resources:	❖ http://chortle.ccsu.edu/Java5/index.html#07 ❖ Java Runtime Environment ❖ Java Development Kit

Unit Title	Working With Types of Data	**Length of Unit**	1-2 weeks

Inquiry Questions (Engaging & Debatable)	● How do we express values and text in Java? ● How does a program interact with a user through input and output streams? ● What types of data can Java work with?
Standards	*** See Above
Unit Strands & Concepts	● Data Types, Primitive Data vs. Objects, The Eight Primitive Data Types of Java ● Numeric Data Types, Character and Boolean Data Types ● Variables, Assignment Statements, Expressions, Arithmetic Operators, Review of Expressions ● Integer operators, Floating point operators, Mixed Floating point and Integer Expressions ● Constants, Classes, Objects, Reference variables ● String objects ● Running methods of objects, Methods that create String objects ● Type wrappers, Packages ● Input and Output Streams, The standard I/O streams ● Scanner class, Exceptions, Converting character data to type int ● Converting strings to double ● The Math class, Square root, Trig functions, PI, Not a Number (NaN)
Key Vocabulary	Primitive Data Type, Bit, Byte, Short, Int, Long, Float, Double, Char, Boolean, Object, Literal, Scientific Notation, Variable, Declaration, Nested ( ), Subexpressions, Constant, Calling a method, Dot notation, Package, Wrapper class, Input stream, Output stream, Sqrt(), NaN, Type casting

Unit Title	Working with Types of Data	**Length of Unit**	1-2 weeks

Critical Content: **My students will Know…**	**Key Skills:** **My students will be able to (Do)…**
● There are several primitive data types with different purposes, using varying degrees of memory. ● All data is expressed as either primitive data types or objects. ● Division performed on two ints produces an answer with no remainder. ● The reserved word “final” will tell the compiler not to change the value of a variable. ● A class is a description of a kind of object. It describes what an object of a particular type is made of(data and methods). ● Creating an object reference is not the same as creating an object. ● NaN stands for “Not a Number”.	● Make appropriate declarations of variables, avoiding reserved words. ● Perform operations on numerical variables, including writing simple counting programs. ● Write simple programs that use the scanner class to input data. ● Interpret a number in scientific notation as a really large or really small number. ● Utilize the sqrt() method to take the square root of floating point values. ● Utilize methods of the Math class to solve problems involving trigonometry.

Assessments:	Program writing activities including, but not limited to, math related algorithms and challenges from projecteuler.net 1-2 quizzes to be created and linked to in implementation guide 1 unit test to be created and linked to in implementation guide
Teacher Resources:	http://chortle.ccsu.edu/Java5/index.html#07 Java Runtime Environment Java Development Kit

Unit Title	Branches and Loops	**Length of Unit**	3-4 weeks

Inquiry Questions (Engaging & Debatable)	● Why is recursion beneficial? ● How do we repeat a command in java as many times as we like? ● How can we maintain a procedure in java until a condition is met?
Standards	*** See Above
Unit Strands & Concepts	• Two-way Decisions, The if statement, Outline of a two-way decision, Blocks of statements, Boolean Expressions, Relational Operators, Example Programs, Single-branch if statements • Boolean expressions that use arithmetic, Picking the right relational symbol, Several if statements in a program, Relational Operators (review) • Logical Operators, AND Operator, How to check that a number is in range • Boolean Expressions, OR Operator, Comparison between AND and OR, NOT Operator, The while statement, Syntax and semantics of the while statement, The loop body, Three parts of a loop that must be coordinated, Counting loops and the loop control variable, Counting upward by two's, The while condition as a gatekeeper, Iteration, Decrementing the loop control variable, Infinite loops, Basing a value on the loop control variable • Adding up even and odd integers, Computing N factorial, Dropping a brick, Making a rectangle of stars, Sentinel-controlled loops, Using an if-else statement inside the loop body • Matching up if's with else's, Using a String value as a sentinel, Result-controlled loops, Newton's method, Nested Loops, Improving an estimate by iteration • Precision of double precision variables, Random numbers and generators, Uniform and non-uniform distributions, The Random class, Seeds for Random(), Tossing dice, Picking passwords
Vocabulary	Binary Decision, If Statement, Block Statement, Boolean Expression, Flowchart, Nested, Relational Expression, Relational Operator, Logical Operator, Truth Table, “!” operator, Loop, While Statement, Iteration, Infinite Loop, Nested Loops, Sentinel, Random(),Generator, Pseudorandom, Seed

Unit Title	Branches and Loops	**Length of Unit**	3-4 weeks

Critical Content: My students will Know…	Key Skills: My students will be able to (Do)…
● A boolean expression can take on one of two values: true or false. ● Short circuit evaluation means that sometimes Java only needs to check the first relational expression to evaluate the value of a more complex boolean expression. ● The "!" operator has precedence over other operators. ● Random numbers generated by the Random() method are only pseudorandom generated by a seed value and an algorithm.	● Write if and if-then statements as part of a Java program. ● Construct a flowchart to model the workings of a Java program with multiple branches and loops. ● Use the "&&" and "||" operators to write complex boolean expressions. ● Use the "!" operator to reverse the value of boolean expressions. ● Use "while" loops, with and without iteration in Java programs. ● Use the Random() method of the Random Class to generate random digits for use in Java programs.

Assessments:	Program writing activities including, but not limited to, math related algorithms and challenges from projecteuler.net 3-4 quizzes to be created and linked to in implementation guide 1 unit test to be created and linked to in implementation guide
Teacher Resources:	http://chortle.ccsu.edu/Java5/index.html#07 Java Runtime Environment Java Development Kit

Unit Title	File Input and Output	**Length of Unit**	1-2 weeks

Inquiry Questions (Engaging & Debatable)	● How does a Java program interact with a user? ● What issues (exceptions) arise when seeking input from a user? ● How can we format data so that it is recognizable to a Java program?
Standards	*** See Above
Unit Strands & Concepts	● Idea of a file, Text files, Redirecting output to a text file, Input file redirection ● Problems with blank lines, Redirecting both input and output, Numeric input from a text file ● Problems with illegal characters in a number, The trim() method, Using Scanner with disk files., I/O Exceptions, hasNext Methods of Scanner ● Class PrintStream, print() and println() Methods of PrintStream ● Three types of input loops, Counting input loops ● Several groups of data in one file, Sentinel controlled input loops ● Default Locale, Formatting numeric output, Decimal Format, Changing the Locale ● Format patterns ○ 0 format character ○ . format character ○ , format character ○ # format character ● Formats for positive and negative numbers
Key Vocabulary	File, Text File, File redirection, Output redirection, Standard error output stream, Concatenation, I/O Streams, Scanner, IOException, hasnext(), hasnextint(), print(), println(), Internationalization, Default Locale, format()

Unit Title	File Input and Output	**Length of Unit**	1-2 weeks

Critical Content: My students will Know…	Key Skills: My students will be able to (Do)…
• How input and output is processed within the workings of a Java program. • The format of input can cause issues within a Java program. • The same type of data can be expressed in many different formats.	• Write Java programs that prompt the user for input. • Write Java programs that produce output. • Write Java programs that input to a .txt file. • Format the values of expressions using the format() method.

Assessments:	Program writing activities including, but not limited to, math related algorithms and challenges from projecteuler.net 1-2 quizzes to be created and linked to in implementation guide 1 unit test to be created and linked to in implementation guide
Teacher Resources:	http://chortle.ccsu.edu/Java5/index.html#07 Java Runtime Environment Java Development Kit

Unit Title	Object Oriented Programming	**Length of Unit**	4-5 weeks

Inquiry Questions	● How does Java differ from older programming languages like Basic? ● What is Objected Oriented Programming and what are its advantages?
Standards	*** See Above
Unit Strands & Concepts	• What is an Object?, What is a Class?, Characteristics of Objects, Static Methods, Constructors, Cookie Cutters, Dot Notation, Primitive data types, and objects, Object references and object reference variables, Copying object references, The == operator and object references, The equals() method for strings • String literals, The Point class, Class descriptions, The toString() method, Changing the data in an object vs constructing a new object, The equals() method, The == operator (again), Aliases • Methods with no parameters, Parameter lists, Data types of parameters • Expressions in parameter lists, Conversion of values in parameter lists to required types, String literals, The null value, More about garbage, The String class, String concatenation, Strings are immutable, Cascading methods, Some String methods, Object Oriented Programming, Syntax of Class Definitions, Syntax of Method Definitions, Object Oriented "Hello World" example, Classes for Testing, How classes are documented • Writing a class definition, Instance variables, Using instance variables in a method, Constructors, Constructor parameter lists, Requirements Analysis, Design, Implementation, Scaffolding, Testing, Aliasing (Review), The private and the public visibility modifiers, Default visibility, Encapsulation, Access methods, Encapsulation of the CheckingAccount class, Formal and Actual Parameters, Scope of Parameters, Local Variables, Method Overloading, Signature of a Method, Review of parameters, Call by value with primitive data types, Object references as parameters, Immutable objects as parameters, Changeable objects as parameters, Defining a class that contains smaller classes, Example: Making a Fleet object out of Car objects, Example: Defining Fleet methods in terms of Car methods
Key Vocabulary	Object, Class, Identity, State, Behavior, Instantiation, Dot Notation, Static, Constructor, Immutable, Object Reference Variable, Garbage Collector, String Literal, Point class, toString(), equals(), Alias, Method call, Type cast, The null value, concat(), Overloaded, Parameter, trim(), charat(), substring(), startswith(), Cascade, String Indexing, Return, Void, Default constructor, Private, Public, Encapsulation, Scope, Signature

Unit Title	Object Oriented Programming	**Length of Unit**	4-5 weeks

Critical Content: My students will Know…	**Key Skills: My students will be able to (Do)…**
• An object has identity, state, and behavior. • A class is a description of a possible object whereas an object is a unique instance of a class. • A program can execute a static method without first creating an object. • An object reference is assigned to a variable, not the object itself. • A reference variable that contains no object is “null”. • String objects are immutable. • Private variables and methods may only be used within their assigned class whereas public variables and methods are accessible to all classes in a package. • Objects may contain other objects.	• Create point objects for use in Java programs. • Created new strings through various methods of String objects. • Write class and method definitions to perform various tasks. • Create classes that contain instance variables, constructors, and methods. • Model a car’s gas mileage with a Java program involving multiple classes, objects, and methods. • Model a series of bank accounts with a Java program involving multiple classes, objects, and methods. • Determine whether a variable or method should be declared public or private.

Assessments:	Program writing activities including, but not limited to, math related algorithms and challenges from projecteuler.net Formative and Summative Assessments linked to an implementation guide
Teacher Resources:	http://chortle.ccsu.edu/Java5/index.html#07 Java Runtime Environment Java Development Kit

Unit Title	More Java Features	**Length of Unit**	2-3 weeks

Inquiry Questions (Engaging & Debatable)	• How can we simply some of the more common commands used in Java Programming? • Are there other ways to set up loops? • How can we program non-binary situations?
Standards	*** See Above
Unit Strands & Concepts	• Increment, Decrement and Assignment Operators • Harmonic Series Example, Short-circuit && and || operators, Non-short-circuit & and | operators • Expressions made with several && operators, several || operators and with mixed && and || operators, Precedence of && and || operators • Truth Tables, De Morgan's Rules • The three parts of all loops (review), Syntax of the for statement, Using the for statement in counting loops, Equivalence of for and while loops, Declaring the loop control variable in the for statement, Scope of a loop control variable • Floating point increments, The conditional operator ?, The switch statement, The do statement • Bottom-driven loops, Dangers of bottom-driven loops
Key Vocabulary	Postfix operator, Prefix operator, Short-circuit evaluation, Pure function, Associativity, Truth table, DeMorgan's Rules, For loop, Top-driven loop, Scope, Scope rule, Switch, Conditional operator, Conditional expression, Break, Do statement, Do-while statement, Bottom-driven loops

Unit Title	More Java Features	**Length of Unit**	2-3 weeks

Critical Content: **My students will Know…**	**Key Skills:** **My students will be able to (Do)…**
• Postfix operators increment after use, where prefix operators increment before use. • Truth tables can provide valuable clarity into the possible outcomes from a boolean expression. • Short circuit evaluation means that a program often only has to evaluate part of a boolean expression. • There is a greater potential for programming bugs when using bottom-driven loops.	• Utilize increment operators to simplify programs. • Write For statements as part of a Java program. • Evaluate the truth of a boolean expression using a truth table. • Write Switch statements as part of a Java program. • Write Do and Do-while statements as part of a Java program.

Assessments:	Program writing activities including, but not limited to, math related algorithms and challenges from projecteuler.net Formative and Summative Assessments linked to an implementation guide
Teacher Resources:	http://chortle.ccsu.edu/Java5/index.html#07 Java Runtime Environment Java Development Kit

Unit Title	Graphics and Arrays	**Length of Unit**	3-4 weeks

Inquiry Questions (Engaging & Debatable)	● How can we create visual representations in a Java program? ● Why is using an array better than using a large number of variables? ● What types of data can be stored in an array?
Unit Strands & Standards	*** See Above
Concepts	• The JFrame class, Methods of the JFrame class: • The JPanel class, Methods of the JPanel class: • The Graphics class, Methods of the Graphics class: • Graph paper graphics, Using Graphics methods to draw a picture, Calculating coordinates in a picture, Scaling a picture to match the frame • Class Polygon, drawPoly() and fillPoly(), The idea of arrays, Array declaration, Array declaration and construction, Using arrays, Automatic bounds checking, Initializer lists, Arrays and counting loops • Enhanced for loop, The length of an array, Finding the maximum and minimum of an array of numbers, Finding the sum of the elements of an array, Computing the average of the elements of an array, Parameters of methods (review), Arrays as parameters, Picture of array parameter passing, Primitive types as parameters, Review of parameter passing with int variables, Parameter passing with array reference variables, A method to zero an array, A method to copy one array to another, Arrays of object and string references, Command line parameters, Linear search of an array, Phone directory application, Two-dimensional Arrays, 2D Array Declaration, Initializer Lists, Different Numbers of cells in each row, 2D Arrays as Arrays of 1D Arrays, Printing a 2D Array • Immutable Strings, charAt() method of String, Reversing a string, StringBuffer, Palindromes
Key Vocabulary	Swing, JFrame, JPanel, Pixel, drawline(), drawOval(), drawRect(), setColor(), RGB, fillRect(), fillOval(), Polygon, addPoint(), Array, Cells, Index, Subscript, Element, Subscripted Variable, null, initializer list, one dimensional array, algorithm, off-by-one error, ArrayOps class, findMax(), call by values, print(), copy(), String[] args, , parseInt(), search(), two dimensional array, row, column, bounds checking, charAt()

Unit Title	Graphics and Arrays	**Length of Unit**	3-4 weeks

Critical Content: **My students will Know…**	**Key Skills:** **My students will be able to (Do)…**
• Graphics programming in Java is done using Swing. • An array is an object. • Each cell of a numeric array is initialized to zero; each cell of an array of references is initialized to null. • A statement of the form arrayB=arrayA will only copy the object reference in arrayA to arrayB, it will not change the contents of arrays. • Because an array index starts at 0, but counting loops usually start at 1, there is a high likelihood of error without careful attention to detail. • Arrays can contain string references inside cells.	• Use methods of the JFrame class to draw images as part of a Java Program. • Model the drawing of a house based on a sample graphics example. • Initialize, adjust, and read values from an array in a variety of Java programs. • Use methods of the ArrayOps class to change arrays. • Construct two-dimensional arrays for use in Java programs. • Utilize the StringBuffer class to modify strings in Java programs.

Assessments:	Program writing activities including, but not limited to, math related algorithms and challenges from projecteuler.net Formative and Summative Assessments linked to an Implementation Guide
Teacher Resources:	http://chortle.ccsu.edu/Java5/index.html#07 Java Runtime Environment Java Development Kit

Unit Title	Graphical User Interface Programming	**Length of Unit**	3-4 weeks

Inquiry Questions (Engaging & Debatable)	• How can we create more dynamic input opportunities for users? • What are the most common ways for a program to interact with a user? • What is involved in creating a Graphical User Interface?
Unit Strands & Standards	*** See Above
Concepts	• Event-driven programming, Graphical user interfaces, The three parts of a GUI, GUI components • Containers, The Abstract Windowing Toolkit (AWT), How the AWT relates to Swing, Swing class hierarchy • The JFrame class., Extending the JFrame class, The JLabel class., Adding a button to a frame • Layout managers, Action listeners, The actionPerformed() method, The content pane of a frame, Changing the background color of a frame, A Button's setActionCommand() method, An Action Event's getActionCommand() method, Using actionPerformed() with several buttons, The JTextField Swing component, The setText() and getText() methods of JTextField, The JLabel GUI Component, Using ActionListener for a JTextField's event, Example program: text echo program, The setEditable() method of JTextField, Example: Temperature conversion application, Obtaining numeric values from JTextFields, The Integer.parseInt() method, Writing results into JTextFields, ActionListener for a JTextField, Using a JButton with JTextField input, The Double.parseDouble() method, Finding the right size for a frame, The setResizable() method, Arranging components in a JPanel, Nested JPanels, The BoxLayout layout manager, Radio buttons, Button groups, The BorderLayout layout manager, The Box class, Glue, sliders, Major and minor ticks. The setPreferredSize() methodThe setName() method and getName() method, The getSource() method
Key Vocabulary	Swing classes, event, application method, Component class, JComponent class, JApplet class, JFrame, setSize(), setBounds(), setVisible(), setDefaultCloseOperation(), frame, JButton, IFlowLayout, content pane, actionPerformed(), JTextField, , setText(), setEditable(), Jpanel, JRadioButton, ButtonGroup, JSliders, tick marks, setMajorTickSpacing(), getSource(), getName(), getvalue(),

Unit Title	Graphical User Interface Programming	Length of Unit	3-4 weeks

Critical Content: My students will Know…	Key Skills: My students will be able to (Do)…
• A GUI(Graphical User Interface) is the basis for Event-driven programming. • A container is a GUI component that has other GUI components placed inside of it. • All GUI programs have three parts: Graphical Components, Listener Methods, and Application Methods. • The most common GUI components are buttons, text fields, radio buttons, and sliders. • A Jpanel is a section of the window screen without defined borders.	• Create an object of the JFrame class with specific dimensions, starting position on the screen, visibility, and closing conditions. • Register listener objects to create a responsive GUI. • Create button objects for a Java program • Create text fields for a Java program • Analyze the structure of two sample programs, one to convert temperature and one to calculate %calories from fat. • Create panels as part of a JFrame in Java programs. • Create radio buttons as part of a ButtonGroup for Java program. • Create sliders as part of a Java Program.

Assessments:	Program writing activities including, but not limited to, math related algorithms and challenges from projecteuler.net Formative and Summative Assessments
Teacher Resources:	http://chortle.ccsu.edu/Java5/index.html#07 Java Runtime Environment Java Development Kit