PUBLIC SCHOOLS OF EDISON TOWNSHIP

OFFICE OF CURRICULUM AND INSTRUCTION

Java Programming

Length of Course:	Full Year
Elective/Required:	Elective
Schools:	High School
Eligibility:	Grade 9-12
Credit Value:	5 Credits
Date Approved:	August 17, 2021

TABLE OF CONTENTS

This guide was revised by: Robert Giordano (JPS) and John Krajunus (EHS)

Completed under the supervision of: Nicole Halpin: Supervisor of 21st Century Skills

A copy of this curriculum guide for review on the District website and in the Office of Curriculum and Instruction.

Suggested Time Schedule

Unit Number	Unit Name	Total Days
1	Intro to Java	10
2	Variables and Math	10
3	Output, Input, and Strings	25
4	Control Structures I - Conditional Statements	20
5	Control Structures II - Loops	25
6	Static Methods	25
7	Classes and Objects	40
8	Data Structures - Arrays	25
Total		180

Unit of Study: 1 - Introduction to Comp Sci

Targeted State Standard(s):

- 8.CS.2 - Software and hardware determine a computing system's capability to store and process information. The design or selection of a computing system involves multiple considerations and potential trade-offs. (A)
- 12.CS.2 - A computing System involves interaction between the user, hardware, application software, and system software. (B)
- 8.IC.1 - Advancements in computing technology can change individuals' behaviors. (C)
- 8.IC.2 - Society is faced with trade offs due to the increasing globalization and automation that computing brings.(D)
- 12.IC.1 - The design and use of computing technologies and artifacts can positively or negatively affect equitable access to information and opportunities. (E)
- CS&DTP - 4 Developing and Using Abstractions (F)

Unit Objectives/Enduring Understandings:

- Students will understand the origins of computing, its strengths and weaknesses, and the relationship between hardware, software, and user.

Essential Questions:

- What are the 4 essential components of every computing device?
- What are the pros and cons of the increasing access to technology and information?
- How has the prevalence of computers affected the quality of life of humans and affected our society?
- What are the responsibilities of the programmer in the computer application design process?
- What is abstraction and what is its purpose?

Unit Assessment:

- Computing Impact Project - Research of a Computing Technology and its impact on people's lives (positive and negative).
- Computer Research Project - Research a computer on the market and determine the components that it has to determine its effectiveness as a computer.

	Core Content Objectives		**Instructional Actions**	
Cumulative Progress Indicators	**Concepts** (What students will know)	**Skills** (What students will be able to do)	**Activities/ Strategies** (Technology Implementation/ Interdisciplinary Connections)	**Assessment Check Points**
• Identify the differences between hardware and software and the purpose of each. (B) • Identify the 4 main components of every computing device. (A) • Convert numbers between number systems (binary, decimal, hexadecimal, octal) • Identify the impact of computing on society (C, D, E) • Describe abstraction and give examples of abstraction in everyday life.(F)	• History of Computers • Hardware Vs. Software • Abilities and Impacts of Computing on Society • Number Systems (binary, octal, decimal, hexadecimal) • Setting up a project/package/class in Eclipse.	• Identify the components of a computer • Determine the difference between hardware and software. • Convert numbers between number systems. • Create a class in Eclipse/other IDE • Maintaining an organized digital workspace.	• Powerpoint/Slides presentations • Research Projects • Creating Computational Artifacts	• Reflection questions • Show What You Know Questions. • Workspace check

Resources: Essential Materials, Supplementary Materials, Links to Best Practices

- Textbook and related resources
- MacBook
- Eclipse Java IDE/Similar IDE
- Website articles
- Video Library

Instructional Adjustment: Modifications, student difficulties, possible misunderstandings

- Circulate during work time to answer questions and provide clarity.
- Utilize pair programming to ensure students have a person with whom to collaborate and discuss difficulties.(CS&DTP - 1, 2, & 7)

Unit of Study: 2 - Variables and Math

Targeted State Standard(s):
- 8.AP.2 - Programmers create variables to store data values of different types and perform appropriate operations on their values.

Unit Objectives/Enduring Understandings:
- Students will understand the different types of storage that Java utilizes, how to create variables that exist in that storage, and how to access their values.
- Students will understand how to use the Math and Random objects and their methods.

Essential Questions:
- What are the 2 types of storage Java uses?
- What are 2 ways to generate a random number?
- What 2 components are required to declare a variable?
- What is typecasting?

Unit Assessment:
- Data Type and Variable Declaration Poster Project

	Core Content Objectives		Instructional Actions	
Cumulative Progress Indicators	**Concepts** (What students will know)	**Skills** (What students will be able to do)	**Activities/ Strategies** (Technology Implementation/ Interdisciplinary Connections)	**Assessment Check Points**
• Identify the different types of data that a computer can work with (data types) • Identify the types of storage used by Java • Declare a variable and, initialize, assign, and access its value. • Perform calculations using the value of a variable. • Use the Math object methods to access higher functionality than arithmetic operators • Use the Random object to generate random numbers and store them in variables.	• Storage in Java • Primitive Data Types. • Identifier names • Declaring and Initializing/Assigning a Variable • Accessing the Value of a Variable • Arithmetic Operators, Order of Operator Precedence, and the Math Object Methods • Generating a Random Number using the Random and Math Objects	• Declare, initialize, assign, and access the value of variables. • Typecast data to another data type. • Perform arithmetic operations on variables. • Use variables to program formulas • Use math methods to perform advanced functions (powers, roots, trig, etc) • Generate random numbers	• Powerpoint/Slides presentations • Poster Project (Make a poster about a data type) • Connection to Mathematics/Sciences for formulas to program. • Discovery Activities • Collaborative/Pair Programming (12.CS.3, 8.DA.5)	• Reflection Questions • Show What You Know Questions • Workspace check

Resources: Essential Materials, Supplementary Materials, Links to Best Practices
- Textbook and Related Resources
- MacBook
- Poster Board and Coloring Supplies
- Eclipse/Similar IDE
- Video Library

Instructional Adjustment: Modifications, student difficulties, possible misunderstandings
- Circulate during work time to answer questions and provide clarity.
- Utilize pair programming to ensure students have the ability to discuss their issues with a peer before asking in front of the class. . (CS&DTP - 1, 2, & 7)

Unit of Study: 3 - Output, Input, and Strings

Targeted State Standard(s):

- 8.DA.3 - Data is represented in many formats. Software tools translate the low-level representation of bits into a form understandable by individuals. Data is organized and accessible based on the application used to store it. (A)
- 12.CS.2 - A computing system involves interaction among the user, hardware, application software, and system software.(B)

Unit Objectives/Enduring Understandings:

- Students will use the Scanner object to take input from the user through the console.
- Students will use the Decimal Format object to format the output of numbers with a decimal component.
- Students will use the print/println statement to write text to the console.

Essential Questions:

- What is the difference between print and println?
- How do we create a scanner object?
- Which scanner method do we use for each type of input data ?
- How do we format a number using the Decimal Format object?
- What are some of the issues that we can experience when using concatenation?

Unit Assessment:

- Programming Project using output, input, variables, and the math and the decimal format objects.

	Core Content Objectives		**Instructional Actions**	
Cumulative Progress Indicators	**Concepts** (What students will know)	**Skills** (What students will be able to do)	**Activities/ Strategies** (Technology Implementation/ Interdisciplinary Connections)	**Assessment Check Points**
• Using output from the program to interact with a user (B) • Using computer storage to hold complex strings of text (A) • Using built in sequences that have special purposes (A) • Using input to allow the user to interact with the program. (A & B)	• The difference between print and println statements • How the computer treats the ' + ' operator when used for concatenation • How computers store text • How to use String methods • How to use escape sequences • How to format a decimal number for printing • How to take input from the user.	• Use the print/println statement to output text to the console. • Concatenate strings both inside and outside of a print/println statement • Store text in a variable and use String methods • Format a decimal number • Use the Scanner class to take in input from the user.	• Powerpoint/Slides presentations • Discovery Activities • Math/Science Formulas • Collaborative/Pair Programming (12.CS.3,8.DA.5) • Passion Projects(12.CS.3, 8.DA.5, 8.AP.5, 12.ED.1)	• Reflection Questions • Show What You Know Questions • Classwork programming assignments • Workspace check

Resources: Essential Materials, Supplementary Materials, Links to Best Practices

- Textbook and Related Resources
- MacBook
- Eclipse/Similar IDE
- Video Library

Instructional Adjustment: Modifications, student difficulties, possible misunderstandings

- Circulate during work time to answer questions and provide clarity.
- Utilize pair programming to ensure students have a person with whom to collaborate and discuss difficulties. (CS&DTP - 1, 2, & 7)

Unit of Study: 4 - Control Structures I - Conditional Statements

Targeted State Standard(s):
- 8.AP.3 - Control structures are selected and combined in programs to solve more complex problems. (A)
- 12.AP.3 - Trade-offs related to implementation, readability, and program performance are considered when selecting and combining control structures.(B)
- 12.AP.1 - Individuals evaluate and select algorithms based on performance, reusability, and ease of implementation.(C)

Unit Objectives/Enduring Understandings:
- Students will use conditional statements to alter the flow of their programs
- Students will choose the "best" conditional statement for the task.

Essential Questions:
- What is the purpose of the "else" statement?
- What is the proper structure of an else-if chain?
- What data types can be used with a switch statement?
- When would you use a switch statement over an if statement and vice versa?

Unit Assessment:
- Programming Project(s) using previous unit content and conditional statements.

	Core Content Objectives		**Instructional Actions**	
Cumulative Progress Indicators	**Concepts** (What students will know)	**Skills** (What students will be able to do)	**Activities/ Strategies** (Technology Implementation/ Interdisciplinary Connections)	**Assessment Check Points**
• Using an if statement to execute code only when a given condition is satisfied.(A) • Using the else statement and else if statement to expand the number of outcomes to model a scenario.(A) • Use the switch statement to program multiple outcomes.(A) • Using the default statement to act as an "else" statement in a switch statement.(A) • Properly nesting if statements to accurately reflect a situation/scenario. (A) • Choosing the right conditional statement for the task (B & C)	• How to program an if statement to execute code when its condition is true. • How to use the else statement to add code that executes if the condition of an if statement is false • How to write an if/else chain. • How to program a switch statement to model a situation with multiple scenarios. • When to use an if/else chain and when to use switch, • How to design a flowchart algorithm. • How to determine the efficiency of an algorithm	• Evaluate complex boolean logic statements to determine their truth value. • Use comparison and logical operators to restrict the bounds of a range of numbers. • Use if statements to alter the flow of the program code. • Use the else and else if statements to more accurately model real-world scenarios • Nest conditional statements. • Use the switch statement to control the flow of the program. • Design an algorithm to accomplish a task	• Powerpoint/Slides Presentations • Discovery Activities • Create Your Own Adventure Project • Flowchart Algorithms • Collaborative/Pair Programming (12.CS.3, 8.DA.5) • Passion Projects(12.CS.3, 8.DA.5, 8.AP.5, 12.ED.1)	• Reflection Questions • Show What You Know Questions • Classwork programming assignments • Workspace check
Resources: Essential Materials, Supplementary Materials, Links to Best Practices • Textbook and Related Resources • MacBook • Eclipse/Similar IDE • Video Library			**Instructional Adjustment:** Modifications, student difficulties, possible misunderstandings • Circulate during work time to answer questions and provide clarity. • Utilize pair programming to ensure students have a person with whom to collaborate and discuss difficulties. (CS&DTP - 1, 2, & 7)	

Unit of Study: 5 - Control Structures II - Loops

Targeted State Standard:

- 8.AP.3 - Control structures are selected and combined in programs to solve more complex problems. (A)
- 12.AP.3 - Trade-offs related to implementation, readability, and program performance are considered when selecting and combining control structures.(B)
- 12.AP.1 - Individuals evaluate and select algorithms based on performance, reusability, and ease of implementation.(C)

Unit Objectives/Enduring Understandings:

- The difference between how pretest and posttest loops operate.
- The difference between fixed repetition loops and variable repetition loops
- The format of for, while, and do-while loops.
- Which loop structure to use depending on the situation/task.

Essential Questions:

- What is the difference between the for loop and the while and do while loops?
- How do pretest and posttest loops operate?
- What is the difference between fixed and variable length looping?
  How do you decide which loop structure to use?

Unit Assessment:

- Programming projects that focus on previous knowledge being integrated into loops.

	Core Content Objectives		Instructional Actions	
Cumulative Progress Indicators	**Concepts** (What students will know)	**Skills** (What students will be able to do)	**Activities/ Strategies** (Technology Implementation/ Interdisciplinary Connections)	**Assessment Check Points**
• Use for loops to repeat certain lines of code (A) • Use while loops to repeat certain lines of code (A) • Use do-while loops to repeat certain lines of code (A) • Nest looping structures • Choose the correct loop structure for the task (B & C)	• For, While, and Do-While Loop structures. • Nesting Loops • How to determine the efficiency of an algorithm • The difference between pretest and posttest loops • The difference between fixed and variable repetition • How to draw loops on a flowchart • How to describe the time complexity of an algorithm.	• Use for, while, and do-while loops to repeat lines of code in a program. • Nest loops for more complex looping. • Determine the efficiency of a looping algorithm • Choose the correct loop structure for the task. • Evaluate the time complexity of a nested loop structure.	• Powerpoint/Slides Presentations • Discovery Activities • Flowchart Algorithms • Collaborative/Pair Programming (12.CS.3, 8.DA.5) • Passion Projects(12.CS.3, 8.DA.5, 8.AP.5, 12.ED.1)	• Reflection Questions • Show What You Know Questions • Classwork programming assignments • Workspace check

Resources: Essential Materials, Supplementary Materials, Links to Best Practices

- Textbook and Related Resources
- MacBook
- Eclipse/Similar IDE
- Video Library

Instructional Adjustment: Modifications, student difficulties, possible misunderstandings

- Circulate during work time to answer questions and provide clarity.
- Utilize pair programming to ensure students have a person with whom to collaborate and discuss difficulties.(CS&DTP - 1, 2, & 7)

Unit of Study: 6 - Static Methods

Targeted State Standard:

- 8.AP.4 - Programs use procedures to organize code and hide implementation details. Procedures can be repurposed in new programs. Defining parameters for procedures can generalize behavior and increase reusability(A)
- 12.AP.4 - Complex programs are designed as systems of interacting modules, each with a specific role, coordinating for a common overall purpose. Modules allow for better management of complex tasks.(B)
- 8.AP.1- Individuals design algorithms that are reusable in many situations. (C)
- 12.AP.1 - Individuals evaluate and select algorithms based on performance, reusability, and ease of implementation. (D)

Unit Objectives/Enduring Understandings:

- Methods are blocks of code that perform a task.
- Parameters are used to send values to a method.
- Methods can return a single value to the point where it was called, if it is a non-void method.
- Methods are used to abstract unnecessary detail from and avoid repetitive code in the main method..

Essential Questions:

- How does calling a method affect the flow of a program?
- What is the correct way to format a method header?
- How is recursion used in methods to solve problems?
- Why do we use methods?
- How do methods prevent repetition in our code?

Unit Assessment:

- Programming Project using previous topics and multiple methods.

	Core Content Objectives		**Instructional Actions**	
Cumulative Progress Indicators	**Concepts** (What students will know)	**Skills** (What students will be able to do)	**Activities/ Strategies** (Technology Implementation/ Interdisciplinary Connections)	**Assessment Check Points**
• Use a method to perform repetitive actions in a program.(B, C, & D) • Use methods to abstract/hide details from the main method. (A) • Use parameters to pass values to a method.	• Void Static Methods • Value-Returning Static Methods • Method Overloading • Recursion in Methods	• Write methods to perform tasks. • Use parameters to send values to a method • Use the “return” statement to send values back to the point of call. • Use recursion to solve problems. • Choose the correct method type to solve a given problem/situation. • Use Methods to abstract away details from the main method. • Design an algorithm using methods.	• Powerpoint/Slides Presentations • Discovery Activities • Flowchart Algorithms • Collaborative/Pair Programming (12.CS.3, 8.DA.5) • Passion Projects(12.CS.3, 8.DA.5, 8.AP.5, 12.ED.1)	• Reflection Questions • Show What You Know Questions • Classwork programming assignments • Workspace check

Resources: Essential Materials, Supplementary Materials, Links to Best Practices

- Textbook and Related Resources
- MacBook
- Eclipse/Similar IDE
- Video Library

Instructional Adjustment: Modifications, student difficulties, possible misunderstandings

- Circulate during work time to answer questions and provide clarity.
- Utilize pair programming to ensure students have a person with whom to collaborate and discuss difficulties. (CS&DTP - 1, 2, & 7)

Unit of Study: 7 - Classes and Objects

Targeted State Standard:

- 12.AP.4 - Complex programs are designed as systems of interacting modules, each with a specific role, coordinating for a common overall purpose. Modules allow for better management of complex tasks.(A)
- 12.AP.5 - Complex programs are developed, tested, and analyzed by teams drawing on the members' diverse strengths using a variety of resources, libraries, and tools. (B)

Unit Objectives/Enduring Understandings:

- Classes are the blueprints for representing a real world object by describing its state(data) and behavior(methods).
- Objects are instances of a class.
- Objects are stored as references and their reference is passed to methods as parameters.
- A constructor method is used to determine how an object is initialized.
- Objects can be used to provide abstraction for programs.

Essential Questions:

- What is a class and what is an object?
- How are objects stored in Java?
- How are objects initialized?
- Can an object have more than one constructor method?
- How are objects able to abstract details?
- What is encapsulation and why is it important?
- What is the difference between static and instance data/methods?
- What happens when you pass an object to a method as a parameter?

Unit Assessment:

- Programming Project using previous topics and classes and objects.

	Core Content Objectives		**Instructional Actions**	
Cumulative Progress Indicators	**Concepts** (What students will know)	**Skills** (What students will be able to do)	**Activities/ Strategies** (Technology Implementation/ Interdisciplinary Connections)	**Assessment Check Points**
• Write an object class and instantiate an object in an executable class. • Write constructor method(s) to initialize an object using parameters • Use encapsulation to hide and protect an object's data	• How to create a class and instantiate an object from it. • How to define an object's state (data) and behavior (methods) and constructor. • What encapsulation is and why it is important. • How to use encapsulation to protect and hide data. • How to use method overloading on objects. • The difference between static and instance data/methods.	• Create classes to model real world objects • Define the state and behavior of an object class. • Instantiate objects in a program and use their state and behavior to simulate an object. • Use encapsulation to hide and protect data. • Use classes and objects to abstract away details. • Design a class with an object as instance data.	• Powerpoint/Slides Presentations • Discovery Activities • Flowchart Algorithms • Collaborative/Pair Programming (12.CS.3, 8.DA.5) • Passion Projects(12.CS.3, 8.DA.5, 8.AP.5, 12.ED.1)	• Reflection Questions • Show What You Know Questions • Classwork programming assignments • Workspace check

Resources: Essential Materials, Supplementary Materials, Links to Best Practices

- Textbook and Related Resources
- MacBook
- Eclipse/Similar IDE
- Video Library

Instructional Adjustment: Modifications, student difficulties, possible misunderstandings

- Circulate during work time to answer questions and provide clarity.
- Utilize pair programming to ensure students have a person with whom to collaborate and discuss difficulties. (CS&DTP - 1, 2, & 7)

Unit of Study: 8 - Data Structures - Arrays

Targeted State Standard:
- 12.AP.2 - Programmers choose data structures to manage program complexity based on functionality, storage, and performance trade-offs.

Unit Objectives/Enduring Understandings:
- Declaring and initializing an array.
- Arrays are a type of object.
- Arrays are a data structure that holds multiple of the same type of data.
- Arrays can have more than 1 dimension.
- Arrays can hold both primitive data and objects.

Essential Questions:
- How do we use loops to traverse an array without going out of bounds?
- What types of data can an array store?
- What happens when you pass an array as a parameter?
- How do we traverse multidimensional arrays?

Unit Assessment:
- Programming Project using all previous topics and Arrays (Hangman/Tic-Tac-Toe)

	Core Content Objectives		**Instructional Actions**	
Cumulative Progress Indicators	**Concepts** (What students will know)	**Skills** (What students will be able to do)	**Activities/ Strategies** (Technology Implementation/ Interdisciplinary Connections)	**Assessment Check Points**
• Declare and initialize an array using individual assignment and array notation. • Write a loop to traverse through an array. • Pass an array as a parameter to a method. • Create an array to hold multiple objects • Create an array with 2 dimensions.	• Declaring and Initializing an Array • Filling an array • Array sorting algorithms • Traversing through an array. • Arrays are a type of object. • Passing arrays as parameters. • Multi-dimensional arrays • Arrays of object references.	• Create a 1 dimensional array. • Use arrays to store a dataset • Sort a dataset stored in an array. • Perform statistical analysis on a dataset in an array. • Create an array of object references. • Create and fill arrays of 2 dimensions.	• Powerpoint/Slides Presentations • Discovery Activities • Flowchart Algorithms • Collaborative/Pair Programming (12.CS.3, 8.DA.5) • Passion Projects(12.CS.3, 8.DA.5, 8.AP.5, 12.ED.1)	• Reflection Questions • Show What You Know Questions • Classwork programming assignments • Workspace check

Resources: Essential Materials, Supplementary Materials, Links to Best Practices
- Textbook and Related Resources
- MacBook
- Eclipse/Similar IDE
- Video Library

Instructional Adjustment: Modifications, student difficulties, possible misunderstandings
- Circulate during work time to answer questions and provide clarity.
- Utilize pair programming to ensure students have a person with whom to collaborate and discuss difficulties. (CS&DTP - 1, 2, & 7)

Course Content, Requirements, and Evaluation Process

I - **Course Content** - This course will contain the following units of study:

1. Basics of computing, the parts of a computer, impact of computing on society, setting up and maintaining an organized digital workspace, fundamental structure of a Java program, and abstraction.
2. Basics of storage in Java, identifiers, data types, typecasting, declaring a variable and initializing/assigning a value to it, arithmetic operators and the order of operator precedence, methods of the Math object, generating a random number using the Math and Random object classes.
3. Printing to the console, concatenation of strings, strings as a type of data, string methods, escape sequences, formatting numbers with a decimal for printing, and taking in data from the user using the Scanner object and java.util.Scanner.
4. Conditional statements that allow the program to branch, designing a program to have a branching structure, writing algorithms that include the ability for code to be skipped, if/else chains, nested conditionals, and switch statements.
5. For loops, writing algorithms to have repetition built in, nesting loops for more complex operations, time complexity of nested loops, variable length loops (while and do/while).
6. Static methods, declaring and calling a user defined method, void functions vs value returning functions, method overloading, recursion.
7. State and behavior of an object, writing a constructor method, overloading constructor methods, instantiating user defined objects, the difference between an executable class and an object class, creating objects with objects for instance data, passing object references to methods.
8. Declaring and initializing an array, sorting arrays, searching and traversing an array, arrays of objects, declaring and initializing multidimensional arrays.

II - Course Requirements - To complete this course successfully, students will be required to demonstrate a satisfactory (or higher) level of proficiency in:

1. Recognize the importance of computing and its positive and negative impacts on our world.
2. Utilize variables to store information to be used in a program.
3. Utilize input and output to make a program interactive for a user.
4. Use control structures to control the flow of a program.
5. Use methods to perform tasks, avoid repetition, and abstract unnecessary details.
6. Design and write object classes to model real world objects and use objects to simulate events.
7. Utilize a data structure to store multiple pieces of information simultaneously and perform operations on the data stored.
8. Write and design algorithms to solve programming problems.

III - Evaluation Process - Throughout the length of this course, students will be evaluated on the basis of:

1. Classwork/Homework Programming Assignments
2. Programming Projects
3. Maintaining an organized digital workspace
4. Class participation
5. Pair programming