NEPTUNE TOWNSHIP SCHOOL DISTRICT

Computer Science I

Curriculum

Grades 10-12

NEPTUNE TOWNSHIP SCHOOL DISTRICT
Office of the Superintendent
60 Neptune Blvd.
Neptune, NJ 07753-4836

May 31, 2023 Document C1#1

NEPTUNE TOWNSHIP BOARD OF EDUCATION

Chanta L. Jackson, President
Donna L. Puryear, Vice President

Dorothea L. Fernandez	Dianna A. Harris
Jerome M. Hubbard	Shelia B. Jones
Jessie Thompson	Kym Hoffman
April Morgan	Drisana Lashley, Neptune City Rep.

SCHOOL DISTRICT ADMINISTRATION

Tami R. Crader, Ed.D.
Superintendent of Schools

Matthew Gristina, Ed.D.
Assistant Superintendent of Schools

Peter J. Leonard
Business Administrator/Board Secretary

Rosemary Della Sala
Assistant Business Administrator/Assistant Board Secretary

Sally A. Millaway, Ed.D.
Director for Curriculum, Instruction & Assessment

Kathleen Skelton
Director of Special Services

Omar Beltran
Director of School Counseling and Social Emotional Support Services

Lakeda Demery-Alston
Supervisor of Humanities & ESL

Stacie Ferrara, Ed.D.
Supervisor of STEM

Charles Kolinofsky
Supervisor of Data & Information

Meghan Plevier, Ed.D.
Supervisor of Early Childhood Education

ELEMENTARY SCHOOL ADMINISTRATION

Principals

Kathleen Thomsen, Gables
James M. Nulle, Green Grove
Mark K. Alfone, Ed.D., Midtown Community
Joshua Loveland, Shark River Hills
Jerard L. Terrell, Ed.D., Summerfield

MIDDLE SCHOOL ADMINISTRATION

Janelle Opoku, Ed.D., Principal
Thomas Decker, Vice Principal

HIGH SCHOOL ADMINISTRATION

Arlene M. Rogo, Ed.D., Principal
Titania M. Hawkins, Ed.D., Vice Principal
Mahon Ryan-Hannaway, Vice Principal
Patrizia Weber, Vice Principal
Richard Arnao, Administrator for Athletic & Co-Curricular Activities

DEPARTMENT CHAIRPERSONS

Kelly Baldino
Dolores Dalelio
Dawn Reinhardt
Nicole Sanyigo
Megan Tenery
Karen Watt

NEPTUNE TOWNSHIP SCHOOL DISTRICT

COMPUTER SCIENCE I
GRADES 10-12
CURRICULUM

Table of Contents

Curriculum

NEPTUNE TOWNSHIP SCHOOL DISTRICT

Computer Science I

Acknowledgements

The Computer Science I course for Grades 10-12 was developed through the dedicated efforts of Donna Kossey, Neptune High School teacher, with guidance of the district's curriculum steering committee members including Lori Dalelio, Department Chairperson, Dawn Reinhardt, Department Chairperson, Stacie Ferrara,Ed.D., STEM Supervisor and Sally A. Millaway, Ed.D., Director for Curriculum, Instruction and Assessment.

This curriculum guide was developed to build a basic understanding of computer programming. Students will gain confidence in using this skill for both personal and work related purposes. This curriculum was written in alignment with the 2020 New Jersey Student Learning Standards for Computer Science and Design Thinking Standards, 2020 Career Readiness, Life Literacies, and Key Skills and focuses on the skills necessary to familiarize students with computer programming and its applications in business, education, science and industry. It is our hope that this curriculum will serve as a valuable resource for the staff members who teach this course and that they will provide feedback and make recommendations for improvement.

NEPTUNE TOWNSHIP SCHOOL DISTRICT

DISTRICT MISSION STATEMENT

The primary mission of the Neptune Township School District is to prepare all of our students for a life-long learning process and to become confident, competent, socially, and culturally- conscious citizens in a complex and diverse world. It is with high expectations that our schools foster:

• A strong foundation in academic and modern technologies.

• A positive, equitable, and varied approach to teaching and learning.

• An emphasis on critical thinking skills and problem-solving techniques.

• A respect for and an appreciation of our world, its resources, and its diverse people.

• A sense of responsibility, good citizenship, and accountability.

• An involvement by the parents and the community in the learning process.

Neptune Township School District

Educational Outcome Goals

The students in the Neptune Township schools will become life-long learners and will:

- Become fluent readers, writers, speakers, listeners, and viewers with comprehension and critical thinking skills.
- Acquire the mathematical skills, understandings, and attitudes that are needed to be successful in their careers and everyday life.
- Understand fundamental scientific principles, develop critical thinking skills, and demonstrate safe practices, skepticism, and open-mindedness when collecting, analyzing, and interpreting information.
- Become technologically literate.
- Demonstrate proficiency in all New Jersey Student Learning Standards (NJSLS).
- Develop the ability to understand their world and to have an appreciation for the heritage of America with a high degree of literacy in civics, history, economics and geography.
- Develop a respect for different cultures and demonstrate trustworthiness, responsibility, fairness, caring, and citizenship.
- Become culturally literate by being aware of the historical, societal, and multicultural aspects and implications of the arts.
- Demonstrate skills in decision-making, goal setting, and effective communication, with a focus on character development.
- Understand and practice the skills of family living, health, wellness and safety for their physical, mental, emotional, and social development.
- Develop consumer, family, and life skills necessary to be a functioning member of society.
- Develop the ability to be creative, inventive decision-makers with skills in communicating ideas, thoughts and feelings.
- Develop career awareness and essential technical and workplace readiness skills, which are significant to many aspects of life and work.

COMPUTER SCIENCE I

COURSE DESCRIPTION

(5 Credits)

In this introductory course, students will learn and practice computer science terminology and principles using the Java programming language. Fundamentals of Java language syntax and object-oriented programming will be introduced. Students will gain practical experience and apply their programming skills by designing, coding, executing, and debugging simple Java applications. At the completion of the course students will be able to apply Java and object-oriented basics to develop intermediate programs independently, as well as apply their newly learned skills in future computer science courses.

Prerequisite: Successful completion of Algebra I

INTEGRATED SOCIAL AND EMOTIONAL LEARNING COMPETENCIES *The following social and emotional competencies are integrated in this curriculum:*	
Self-Awareness	
x	Recognize one's own feelings and thoughts
x	Recognize the impact of one's feelings and thoughts on one's own behavior
x	Recognize one's personal traits, strengths and limitations
x	Recognize the importance of self-confidence in handling daily tasks and challenges
Self-Management	
x	Understand and practice strategies for managing one's own emotions, thoughts and behaviors
x	Recognize the skills needed to establish and achieve personal and educational goals
x	Identify and apply ways to persevere or overcome barriers through alternative methods to achieve one's goals
Social Awareness	
x	Recognize and identify the thoughts, feelings, and perspectives of others
x	Demonstrate an awareness of the differences among individuals, groups, and others' cultural backgrounds
x	Demonstrate an understanding of the need for mutual respect when viewpoints differ
x	Demonstrate an awareness of the expectations for social interactions in a variety of setting
Responsible Decision Making	
x	Develop, implement and model effective problem solving and critical thinking skill
x	Identify the consequences associated with one's action in order to make constructive choices
x	Evaluate personal, ethical, safety and civic impact of decisions
Relationship Skills	
x	Establish and maintain healthy relationships
x	Utilize positive communication and social skills to interact effectively with others
	Identify ways to resist inappropriate social pressure
x	Demonstrate the ability to present and resolve interpersonal conflicts in constructive ways
x	Identify who, when, where, or how to seek help for oneself or others when needed

Unit Plan Title	Unit 1: Introduction to Computer Programming
Suggested Time Frame	2 weeks

Overview / Rationale

Students will explore the basic foundations of programming, gaining a familiarity with necessary terminology used in computer science and fundamentals of how a program works. They will learn how to write and run small programs using strings, variables, data types, elementary data structures, operators and compositions of statements and expressions in Java.

Stage 1 – Desired Results

Established Goals:

New Jersey Student Learning Standards in Computer Design (2020)

8.1 Computer Science by the End of Grade 12

8.1.12.CS.1: Describe ways in which integrated systems hide underlying implementation details to simplify user experiences.
8.1.12.CS.2: Model interactions between application software, system software, and hardware.
8.1.12.CS.3: Compare the functions of application software, system software, and hardware.
8.1.12.CS.4: Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.
8.1.12.DA.2: Describe the trade-offs in how and where data is organized and stored.
8.1.8.AP.1: Design and illustrate algorithms that solve complex problems using flowcharts and/or pseudocode.
8.1.12.AP.2: Create generalized computational solutions using collections instead of repeatedly using simple variables.
8.1.12.AP.3: Select and combine control structures for a specific application based upon performance and readability, and identify trade-offs to justify the choice.
8.1.12.AP.5: Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.
8.1.12.AP.6: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.
8.1.12.AP.8: Evaluate and refine computational artifacts to make them more usable and accessible.

New Jersey Student Learning Standards in Design Thinking (2020)

8.2 Design Thinking by the End of Grade 12

8.2.12.ED.5: Evaluate the effectiveness of a product or system based on factors that are related to its requirements, specifications, and constraints (e.g., safety, reliability, economic considerations, quality control, environmental concerns, manufacturability, maintenance and repair, ergonomics).
8.2.12.NT.1: Explain how different groups can contribute to the overall design of a product.

Essential Questions:

- What is a program and how is it important in computer science?
- How are variables used in programs and why do data types matter?
- What are the three types of software bugs? How are they different? How are they discovered and resolved?

Enduring Understandings:
Students will understand that…

- A program is a sequence of instructions that specifies how to perform a computation.
- A variable is a named location that can be printed, stored and operated on. A type is associated with a variable through a declaration.
- A syntax error, logic error, or run-time error can occur in a program, and these bugs can be discovered and resolved using a variety of debugging techniques.

Knowledge:
Students will know…

- Java is a programming language used to program computers to perform functions.
- Variables in Java are assigned a type and their values can be modified.
- Strings and variables are used in program input and can be used to create various formats of output.
- Keywords, operators, and operations are predefined in Java and have a special meaning.
- Statements and expressions can be combined to form more complex expressions in Java.

Skills:
Students will be able to…

- Design, write and execute small Java programs.
- Demonstrate the use of variables, variable assignment and variable type definition.
- Utilize string and variable input/output.
- Utilize keywords, operators and operations.
- Design and use compositions of statements/expressions in a program.

Interdisciplinary Connections

New Jersey Student Learning Standards for English Language Arts (2016)
NJSLSA.W8. Gather relevant information from multiple print and digital sources, assess the credibility and accuracy of each source, and integrate the information while avoiding plagiarism.

Standards for Mathematical Practices (2016)
1. Make sense of problems and persevere in solving them.
2. Reason abstractly and quantitatively.
3. Construct viable arguments and critique the reasoning of others.
4. Model with mathematics.
5. Use appropriate tools strategically.
6. Attend to precision.
7. Look for and make use of structure.
8. Look for and express regularity in repeated reasoning.

NJSLS Career Readiness,Life Literacies, and Key Skills (2020)
9.1.12.FP.5: Evaluate how behavioral bias (e.g., overconfidence, confirmation, recency, loss aversion, etc.) affects decision-making.

9.2.12.CAP.5: Assess and modify a personal plan to support current interests and postsecondary plans.
9.2.12.CAP.6: Identify transferable skills in career choices and design alternative career plans based on those skills.
9.4.12.CI.1: Demonstrate the ability to reflect, analyze, and use creative skills and ideas.
9.4.12.CI.2: Identify career pathways that highlight personal talents, skills, and abilities.
Critical Thinking and Problem-Solving
9.4.12.CT.1: Identify problem-solving strategies used in the development of an innovative product or practice (e.g., 1.1.12acc.C1b, 2.2.12.PF.3).
9.4.12.CT.2: Explain the potential benefits of collaborating to enhance critical thinking and problem solving)
9.4.12.TL.1: Assess digital tools based on features such as accessibility options, capacities, and utility for accomplishing a specific task.

Student Resources
Texts: Think Java – How to Think Like a Computer Scientist Version 6.1.0 (2016 free text download) http://greenteapress.com/thinkjava6/thinkjava.pdf **Technology:** Java Integrated Development Environment (Eclipse or DrJava) **Websites:** Help with Java syntax and logic: https://stackoverflow.com https://geeksforgeeks.com https://docs.oracle.com/en/java
Teacher Resources
Texts: Think Java – How to Think Like a Computer Scientist Version 6.1.0 (2016 free text download) http://greenteapress.com/thinkjava6/thinkjava.pdf **Technology:** Java language download. Java Integrated Development Environment (Eclipse or DrJava) **Websites:** https://java/com drjava.org https://stackoverflow.com

https://geeksforgeeks.com
https://docs.oracle.com/en/java

K-12 Universal Legislation

Amistad Law N.J.S.A. 18A 52:16A-88 Every board of education shall incorporate the information regarding the contributions of African Americans to our country in an appropriate place in the curriculum of elementary and secondary school students.

Diversity and Inclusion Law (N.J.S.A. 18A:35-4.36a)
Beginning in the 2021-2022 school year, each school district shall incorporate instruction on diversity and inclusion in an appropriate place in the curriculum of students in grades kindergarten through 12 as part of the district's implementation of the New Jersey Student Learning Standards.

Holocaust Law (N.J.S.A. 18A:35-28) Every board of education shall include instruction on the Holocaust and genocides in an appropriate place in the curriculum of all elementary and secondary school pupils. The instruction shall further emphasize the personal responsibility that each citizen bears to fight racism and hatred whenever and wherever it happens.

LGBT and Disabilities Law (N.J.S.A. 18A:35-4.35) A board of education shall include instruction on the political, economic, and social contributions of persons with disabilities and lesbian, gay, bisexual, and transgender people, in an appropriate place in the curriculum of middle school and high school students as part of the district's implementation of the New Jersey Student Learning Standards. N.J.S.A.18A:35-4.36.

Stage 2 – Assessment Evidence

Formative Assessments:

- Anticipatory and Do Now assignments
- Peer assessments
- Self-Assessment
- Closure assessments
- Informal Observations

Summative Assessments:

- Quizzes
- Java Programs

Performance Task(s):

Become familiar with Java Integrated Development Environment (Dr Java or Eclipse), compile and run a simple HelloWorld program.

Write java programs to

- Print the date in American and European format.

- Convert seconds to hours, minutes, seconds.
- Calculate a marking period grade
- Convert a temperature from Celsius to Fahrenheit.

Stage 3 – Learning Plan

Week 1: Writing a simple Java Program
Lesson #1: Your First Java Program

- Discuss how programming, algorithms and computer science are used to solve problems. Provide real-world examples.
- Show an example of how a programming language is used to write a program in an integrated development environment. Provide examples and model an example.
- Describe how source code is formatted, and how a programming language is executed from source code to output.
- Have students create and run a "Hello World" program in Java, and model basic output formatting.
- Model how to identify and resolve simple software bugs.

Lesson #2: Variables and Assignment

Describe how to declare a variable and how to assign a value to it.

- Demonstrate the use of a state diagram to describe how a variable changes during flow of execution of a program.

Lesson #3: Keywords, Operators, Operations

- Guide students to discover how some expressions with mixed types are automatically converted, potentially leading to run-time errors.
- Model how to use operators to combine variables and constants to perform functions.
- Guide students to perform various operations on strings.

Week 2: Composition of Statements
Lesson #4 Composition of Statements/Expressions

- Demonstrate how to combine expressions to create increasingly complex statements.
- Guide students to discover how rounding errors can sometimes occur unexpectedly and why.
- Model how to discover and debug logic and run-time errors.

Unit Plan Title	Unit 2: Java Methods, I/O, and Conditionals
Suggested Time Frame	4 weeks

Overview / Rationale

Students will define Java methods to perform simple functions. Parameters, void methods, methods with types, and library methods will be used and compared. Multiple methods and recursion, and associated stack diagrams will be investigated to gain an understanding of flow of program execution. Students will use simple input/output methods to get data to/from a Java program. Students will use Java conditionals to manipulate the flow of control of a program based on specified conditions.

Stage 1 – Desired Results

Established Goals:

New Jersey Student Learning Standards in Computer Design (2020)

8.1 Computer Science by the End of Grade 12

8.1.12.CS.1: Describe ways in which integrated systems hide underlying implementation details to simplify user experiences.
8.1.12.CS.2: Model interactions between application software, system software, and hardware.
8.1.12.CS.3: Compare the functions of application software, system software, and hardware.
8.1.12.CS.4: Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.
8.1.12.DA.2: Describe the trade-offs in how and where data is organized and stored.
8.1.12.DA.3: Translate between decimal numbers and binary numbers
8.1.12.DA.4: Explain the relationship between binary numbers and the storage and use of data in a computing device.
8.1.12.DA.6: Create and refine computational models to better represent the relationships among different elements of data collected from a phenomenon or process.
8.1.8.AP.1: Design and illustrate algorithms that solve complex problems using flowcharts and/or pseudocode.
8.1.12.AP.2: Create generalized computational solutions using collections instead of repeatedly using simple variables.
8.1.12.AP.3: Select and combine control structures for a specific application based upon performance and readability, and identify trade-offs to justify the choice.
8.1.12.AP.5: Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.
8.1.12.AP.6: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.
8.1.12.AP.8: Evaluate and refine computational artifacts to make them more usable and accessible.

New Jersey Student Learning Standards in Design Thinking (2020)

8.2 Design Thinking by the End of Grade 12

8.2.12.ED.5: Evaluate the effectiveness of a product or system based on factors that are related to its requirements, specifications, and constraints (e.g., safety, reliability, economic considerations, quality control, environmental concerns, manufacturability, maintenance and repair, ergonomics).
8.2.12.NT.1: Explain how different groups can contribute to the overall design of a product.

Essential Questions:

- How does a void method compare to a typed method?
- How are Java methods used and why are they important?
- What is the purpose of a Java conditional?
- What affects the flow of execution of a program?
- What are the organizational units of a Java program?
- How are organizational units important in object oriented programming?
- What is I/O and why is it useful in a Java program?
- How is I/O performed in Java?
-

Enduring Understandings:

Students will understand that…

- A void method has no return value, and a typed method returns a value of a predefined type.
- A Java conditional allows the flow of execution of a program to branch based on specified conditions.
- A Java program consists of the following organizational units: package, class, method, statement, expression, token.
- Simple Java I/O is performed using the System and Scanner classes.

Knowledge:

Students will know…

- A Java method is used to perform a specified function on the given parameters.
- A typed method returns a value of a specified data type.
- Input/output methods can be designed to allow a program to generate formatted output.
- Conditionals are used to control the flow of execution of a program based on inputs.

Skills:

Students will be able to…

- Design and use a void Java method with parameters.
- Design and use a typed Java method.
- Demonstrate and write Java I/O methods.
- Utilize conditionals and trace program flow of execution of a program.

Interdisciplinary Connections

New Jersey Student Learning Standards for English Language Arts (2016)

NJSLSA.W8. Gather relevant information from multiple print and digital sources, assess the credibility and accuracy of each source, and integrate the information while avoiding plagiarism.

Standards for Mathematical Practices (2016)

1. Make sense of problems and persevere in solving them.
2. Reason abstractly and quantitatively.
3. Construct viable arguments and critique the reasoning of others.

4. Model with mathematics.
5. Use appropriate tools strategically.
6. Attend to precision.
7. Look for and make use of structure.
8. Look for and express regularity in repeated reasoning.

NJSLS Career Readiness,Life Literacies, and Key Skills (2020)
9.1.12.FP.5: Evaluate how behavioral bias (e.g., overconfidence, confirmation, recency, loss aversion, etc.) affects decision-making.
9.2.12.CAP.5: Assess and modify a personal plan to support current interests and postsecondary plans.
9.2.12.CAP.6: Identify transferable skills in career choices and design alternative career plans based on those skills.
9.4.12.CI.1: Demonstrate the ability to reflect, analyze, and use creative skills and ideas.
9.4.12.CI.2: Identify career pathways that highlight personal talents, skills, and abilities.
Critical Thinking and Problem-Solving
9.4.12.CT.1: Identify problem-solving strategies used in the development of an innovative product or practice (e.g., 1.1.12acc.C1b, 2.2.12.PF.3).
9.4.12.CT.2: Explain the potential benefits of collaborating to enhance critical thinking and problem solving)
9.4.12.TL.1: Assess digital tools based on features such as accessibility options, capacities, and utility for accomplishing a specific task.

Student Resources
Texts: Think Java – How to Think Like a Computer Scientist Version 6.1.0 (2016 free text download) http://greenteapress.com/thinkjava6/thinkjava.pdf **Technology:** Java Integrated Development Environment (Eclipse or DrJava) **Websites:** Help with Java syntax and logic: https://stackoverflow.com https://geeksforgeeks.com https://docs.oracle.com/en/java
Teacher Resources
Texts: Think Java – How to Think Like a Computer Scientist Version 6.1.0 (2016 free text download) http://greenteapress.com/thinkjava6/thinkjava.pdf **Technology:** Java language download. Java Integrated Development Environment (Eclipse or DrJava)

Websites:
https://java/com
drjava.org
https://stackoverflow.com
https://geeksforgeeks.com
https://docs.oracle.com/en/java

Stage 2 – Assessment Evidence

Formative Assessments:

- Anticipatory and Do Now assignments
- Peer feedback
- Self-Assessment
- Closure assessments
- Informal Observations

Summative Assessments:

- Quizzes
- Java Programs

Performance Task(s):

Become familiar with Java Integrated Development Environment (Dr Java or Eclipse), compile and run a simple HelloWorld program.

Write java programs to

- Write a "Guess My Number" game. User will input a number between a specified range. Output should include how high or low the user's guess is compared to a randomly generated number in the program.
- Write a BMI calculator. Users will input a height in pounds and weight in inches. Program will convert to meters and kilograms, then calculate BMI.
- Write a simple interactive quiz, 3-5 questions, multiple choice. Use a generic Java method to handle each question in the quiz.

Stage 3 – Learning Plan

Week 1: Java I/O

Lesson #1: System and Scanner Classes

- Demonstrate how to read input data from the keyboard to use in a Java program.
- Guide students in coding I/O statements using the System and Scanner Java library classes.
- Explain the structure of a program and how the package, class, method, statement, expression, and token organizational units are related.
- Guide students to write a program to prompt for input from the keyboard and perform an operation on the input data.
- Demonstrate use of literals in a program along with input data.

Lesson #2: Formatting Output.

- Encourage and guide students to experiment with and compare various output formatting statements on the same and different data types.
- Explain how input characters such as newline can sometimes cause unexpected results in program execution.

Week 2: Void Methods

Lesson #1: Void Methods

- Show how to write a void method to perform a function.
- Explain the purpose of math library methods. Demonstrate why and when are they appropriate to use.

Lesson #2: Parameters and Arguments

- Demonstrate the use of Java methods that take parameters and show how to operate on the parameters to perform a function.
- Show how the same function can be performed with and without a Java method and explain the benefits of each.

Lesson #3: Stack Diagrams

- Describe the flow of execution when a method is called in a Java program.
- Illustrate the state of variables and methods at a particular point in time using a stack diagram.

Week 3: Conditionals

Lesson #1: Relational/Logical Operators and Conditional Statements.

- List and describe the relational operations and the purpose of each.
- List and describe the logical operators. Use logical operators to combine relational statements.
- Learn how to control the flow of execution of a program using conditional statements.

Lesson #2: Chaining/Nesting

- Describe chaining and nesting of conditional statements.
- Understand conditions under which chaining and nesting are useful.
- Model how a flag variable is used.

Lesson #3: Recursion

- Discuss the purpose of a return statement.
- Explain recursion and its significance in the flow of control of a program.
- Draw a stack diagram to illustrate simple recursion.

Lesson #4: Binary Numbers

Describe binary numbers and why they are used in computers.

- Guide students to write a recursive method to display the binary representation of a number.

Week 4: Value Methods

Lesson #1: Return Values/Writing Methods.

- Describe the purpose of a value method. Explain how it is different from a void method and why both are necessary?
- Discuss the purpose of a return value.

- Model and discuss helpful strategies for writing methods.

Lesson #2: Method Composition

- Demonstrate the use of method composition to build new methods using existing methods.
- Discuss functional decomposition and its purpose in facilitating coding and debugging.

Lesson #3: Overloading

- Discuss overloading in the context of operators and methods.
- Explain why method overloading is sometimes useful in a program.
- Model how boolean methods are used in a program.

Unit Plan Title	Unit 3: Loops, Arrays, and Strings
Suggested Time Frame	6 weeks

Overview / Rationale
Students will understand and use iteration, encapsulation and generalization to write more readable and manageable code. They will be introduced to data structures, writing programs using arrays. They will learn how to define and traverse a string data type, using an array index. These concepts will be put together to design and implement more complex Java programs.

Stage 1 – Desired Results

Established Goals:

New Jersey Student Learning Standards in Computer Design (2020)

8.1 Computer Science by the End of Grade 12

8.1.12.CS.1: Describe ways in which integrated systems hide underlying implementation details to simplify user experiences.
8.1.12.CS.2: Model interactions between application software, system software, and hardware.
8.1.12.CS.3: Compare the functions of application software, system software, and hardware.
8.1.12.CS.4: Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.
8.1.12.DA.2: Describe the trade-offs in how and where data is organized and stored.
8.1.8.AP.1: Design and illustrate algorithms that solve complex problems using flowcharts and/or pseudocode.
8.1.12.AP.2: Create generalized computational solutions using collections instead of repeatedly using simple variables.
8.1.12.AP.3: Select and combine control structures for a specific application based upon performance and readability, and identify trade-offs to justify the choice.
8.1.12.AP.5: Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.
8.1.12.AP.6: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.
8.1.12.AP.8: Evaluate and refine computational artifacts to make them more usable and accessible.

New Jersey Student Learning Standards in Design Thinking (2020)

8.2 Design Thinking by the End of Grade 12

8.2.12.ED.5: Evaluate the effectiveness of a product or system based on factors that are related to its requirements, specifications, and constraints (e.g., safety, reliability, economic considerations, quality control, environmental concerns, manufacturability, maintenance and repair, ergonomics).
8.2.12.NT.1: Explain how different groups can contribute to the overall design of a product.
8.2.12.NT.2: Redesign an existing product to improve form or function.

Essential Questions:

- What is iteration and encapsulation?
- Why is it important in writing good code?

- What is a data structure and why is it extremely important in computer science?
- How is the array data structure used?
- How are strings used in programs and how are they represented in Java?

Enduring Understandings:
Students will understand that…

- Encapsulation uses generalized and repeated code and defines it in a method. Iteration can be used to run generalized code repeatedly using a loop.
- A data structure organized data in a program. An array is a data structure used to index large amounts of data items of the same type.
- Strings are used as a data type in Java, and they can be represented as arrays.

Knowledge:
Students will know…

- Iteration and recursion are used to repeat generalized code.
- Arrays are a data structure used to facilitate data manipulation and operations on data.
- Strings are a data type in Java and string operators are provided in Java to manipulate them.

Skills:
Students will be able to…

- Design and write code using iteration/loops, and compare it to recursion.
- Design and write code to manipulate data in an array.
- Utilize strings as a data type and perform operations on them.

Interdisciplinary Connections

New Jersey Student Learning Standards for English Language Arts (2016)
NJSLSA.W8. Gather relevant information from multiple print and digital sources, assess the credibility and accuracy of each source, and integrate the information while avoiding plagiarism.

Standards for Mathematical Practices (2016)
1. Make sense of problems and persevere in solving them.
2. Reason abstractly and quantitatively.
3. Construct viable arguments and critique the reasoning of others.
4. Model with mathematics.
5. Use appropriate tools strategically.
6. Attend to precision.
7. Look for and make use of structure.
8. Look for and express regularity in repeated reasoning.

NJSLS Career Readiness,Life Literacies, and Key Skills (2020)
9.1.12.FP.5: Evaluate how behavioral bias (e.g., overconfidence, confirmation, recency, loss aversion, etc.) affects decision-making.

9.2.12.CAP.5: Assess and modify a personal plan to support current interests and postsecondary plans.
9.2.12.CAP.6: Identify transferable skills in career choices and design alternative career plans based on those skills.
9.4.12.CI.1: Demonstrate the ability to reflect, analyze, and use creative skills and ideas.
9.4.12.CI.2: Identify career pathways that highlight personal talents, skills, and abilities.
Critical Thinking and Problem-Solving
9.4.12.CT.1: Identify problem-solving strategies used in the development of an innovative product or practice (e.g., 1.1.12acc.C1b, 2.2.12.PF.3).
9.4.12.CT.2: Explain the potential benefits of collaborating to enhance critical thinking and problem solving)
9.4.12.TL.1: Assess digital tools based on features such as accessibility options, capacities, and utility for accomplishing a specific task.

Student Resources
Texts: Think Java – How to Think Like a Computer Scientist Version 6.1.0 (2016 free text download) http://greenteapress.com/thinkjava6/thinkjava.pdf **Technology:** Java Integrated Development Environment (Eclipse or DrJava) **Websites:** Help with Java syntax and logic: https://stackoverflow.com https://geeksforgeeks.com https://docs.oracle.com/en/java Lab Resource Page – AP Central | College Board
Teacher Resources
Texts: Think Java – How to Think Like a Computer Scientist Version 6.1.0 (2016 free text download) http://greenteapress.com/thinkjava6/thinkjava.pdf **Technology:** Java language download. Java Integrated Development Environment (Eclipse or DrJava) **Websites:** https://java/com drjava.org https://stackoverflow.com https://geeksforgeeks.com https://docs.oracle.com/en/java Lab Resource Page – AP Central | College Board

Stage 2 – Assessment Evidence

Formative Assessments:

- Anticipatory and Do Now assignments
- Peer feedback
- Self-Assessment
- Closure assessments
- Informal Observations

Summative Assessments:

- Quizzes
- Java Programs

Performance Task(s):

Sieve of Eratosthenes- Use a flowchart to design, and then code the algorithm used by the ancient Greek astronomer Eratosthenes to sift out prime numbers from a list.

Write java programs to

- Simulate an iterative PIN verifier, allowing five attempts before a user is locked out of their account.
- Calculate pi using the Leibnitz series formula.
- Create a word guessing game, where a user guesses one letter at a time with a specified number of incorrect guesses allowed.
- Magpie chatbot lab.

Stage 3 – Learning Plan

Week 1: Loops
Lesson #1 Using Loops In Programs.

- Explain the purpose and use of the while statement.
- Demonstrate use of encapsulation and generalization to write more easily modifiable code.
- Guide students in writing Java programs using for and do-while loops.

Week 2: Arrays
Lesson #2: Using Arrays As Beginning Data Structures.

- Show how to create arrays and explain their purpose in accessing elements.
- Clarify use of iteration in array traversal.
- Discuss applications of arrays.

Week 3-6: Strings
Lesson #3: String and character types.

- Introduce char as a primitive data type and compare it to the String class.
- Discuss Unicode and look at examples for different languages and internal representation of character sets.
- Compare string traversal versus char array traversal.
- Use charAt and length string methods in string traversal.

- Show how Java substring, compareTo, replace, equals, and indexOf string methods are used in string manipulation.
- Model and guide students to explore various methods for formatting strings, using Java methods from the string class online Java documentation.
- Investigate how keywords are significant in chatbots. Apply this knowledge in designing a simple chatbot.

Unit Plan Title	Unit 4: Objects and Classes
Suggested Time Frame	1 week

Overview / Rationale

Students will gain a better understanding of object-oriented programming. Students will learn about attributes and methods of an object, and how an object can be used, for example, as a parameter or return type. Students will define classes that represent objects to more complex tasks. Students will be able to understand and explain the essence of object-oriented programming and compare it to structured programming, Students will use Java classes to design modular programs.

Stage 1 – Desired Results

Established Goals:

New Jersey Student Learning Standards in Computer Design (2020)

8.1 Computer Science by the End of Grade 12

8.1.12.CS.1: Describe ways in which integrated systems hide underlying implementation details to simplify user experiences.
8.1.12.CS.2: Model interactions between application software, system software, and hardware.
8.1.12.CS.3: Compare the functions of application software, system software, and hardware.
8.1.12.CS.4: Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.
8.1.12.DA.2: Describe the trade-offs in how and where data is organized and stored.
8.1.12.DA.3: Translate between decimal numbers and binary numbers
8.1.12.DA.4: Explain the relationship between binary numbers and the storage and use of data in a computing device.
8.1.8.AP.1: Design and illustrate algorithms that solve complex problems using flowcharts and/or pseudocode.
8.1.12.AP.2: Create generalized computational solutions using collections instead of repeatedly using simple variables.
8.1.12.AP.3: Select and combine control structures for a specific application based upon performance and readability, and identify trade-offs to justify the choice.
8.1.12.AP.5: Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.
8.1.12.AP.6: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.
8.1.12.AP.8: Evaluate and refine computational artifacts to make them more usable and accessible.

New Jersey Student Learning Standards in Design Thinking (2020)

8.2 Design Thinking by the End of Grade 12

8.2.12.ED.5: Evaluate the effectiveness of a product or system based on factors that are related to its requirements, specifications, and constraints (e.g., safety, reliability, economic considerations, quality control, environmental concerns, manufacturability, maintenance and repair, ergonomics).
8.2.12.NT.1: Explain how different groups can contribute to the overall design of a product.

Essential Questions:

- What is an object-oriented language and what is the significance of object-oriented programming?
- What is an attribute of an object and why is it important?
- How does an object compare to a class?
- Why is the notion of a class important in object-oriented programming?
- What is data encapsulation and why is it important in program design?

Enduring Understandings:

Students will understand that…

- Java is an example of an object-oriented programming language, which uses objects to define data and associated methods.
- Variables that belong to an object are called attributes.
- A class is similar to a template for objects, specifying the attributes and methods for the object.
- Data encapsulation bundles data and related methods in an object so that the data and methods can be treated as a single unit.

Knowledge:

Students will know…

- A class gives a template for an object. An object is an instance of a class.
- Objects can be used as parameters and return types to Java methods.

Skills:

Students will be able to…

- Understand the difference between an object and a class, and use both in program design.
- Design and write code that uses objects as parameters and return types.

Interdisciplinary Connections

New Jersey Student Learning Standards for English Language Arts (2016)

NJSLSA.W8. Gather relevant information from multiple print and digital sources, assess the credibility and accuracy of each source, and integrate the information while avoiding plagiarism.

Standards for Mathematical Practices (2016)

1. Make sense of problems and persevere in solving them.
2. Reason abstractly and quantitatively.
3. Construct viable arguments and critique the reasoning of others.
4. Model with mathematics.
5. Use appropriate tools strategically.
6. Attend to precision.
7. Look for and make use of structure.
8. Look for and express regularity in repeated reasoning.

NJSLS Career Readiness,Life Literacies, and Key Skills (2020)

9.1.12.FP.5: Evaluate how behavioral bias (e.g., overconfidence, confirmation, recency, loss aversion, etc.) affects decision-making.

9.2.12.CAP.5: Assess and modify a personal plan to support current interests and postsecondary plans.

9.2.12.CAP.6: Identify transferable skills in career choices and design alternative career plans based on those skills.

9.4.12.CI.1: Demonstrate the ability to reflect, analyze, and use creative skills and ideas.

9.4.12.CI.2: Identify career pathways that highlight personal talents, skills, and abilities.

Critical Thinking and Problem-Solving

9.4.12.CT.1: Identify problem-solving strategies used in the development of an innovative product or practice (e.g., 1.1.12acc.C1b, 2.2.12.PF.3).

9.4.12.CT.2: Explain the potential benefits of collaborating to enhance critical thinking and problem solving)

9.4.12.TL.1: Assess digital tools based on features such as accessibility options, capacities, and utility for accomplishing a specified task.

Student Resources

Texts:

Think Java – How to Think Like a Computer Scientist Version 6.1.0 (2016 free text download) http://greenteapress.com/thinkjava6/thinkjava.pdf

Technology:

Java Integrated Development Environment (Eclipse or DrJava)

Websites:

Lab Resource Page – AP Central | College Board

Help with Java syntax and logic:

https://stackoverflow.com

https://geeksforgeeks.com

https://docs.oracle.com/en/java

Teacher Resources

Texts:

Think Java – How to Think Like a Computer Scientist Version 6.1.0 (2016 free text download) http://greenteapress.com/thinkjava6/thinkjava.pdf

Technology:

Java language download.

Java Integrated Development Environment (Eclipse or DrJava)

Websites:

https://java/com

drjava.org

https://stackoverflow.com

https://geeksforgeeks.com

https://docs.oracle.com/en/java

Lab Resource Page – AP Central | College Board

Stage 2 – Assessment Evidence
Formative Assessments: ● Anticipatory and Do Now assignments ● Peer assessments ● Self-Assessment ● Closure assessments ● Informal Observations ***Summative Assessments:*** ● Quizzes ● Java Programs ***Performance Task(s):*** Picture Explorer. Chatbot Lab. Write java programs to - Explore how a color object is used to create custom colors in java.

Stage 3 – Learning Plan
Weeks 1: Objects **Lesson #1 Using Java Objects.** - Develop and explain the point and rectangle objects and their use as parameters and return types. - Explain the purpose of mutable objects and aliasing, and guide in using these features in programs. - Illustrate and model the use of class diagrams in programming hierarchy. **Lesson #2: Using Java Classes.** - Discuss the purpose and use of constructors. - Model and demonstrate Java getters and setters in sample programs. - Model methods for displaying objects in Java programs.

Unit Plan Title	Unit 5: Graphics
Suggested Time Frame	5 weeks

Overview / Rationale

Students will learn how use a Java library package to write programs to create simple 2D graphics. Students will understand how pixels and Java graphical coordinates are used to represent dots on the screen, and use Java library classes to draw lines and colors on the canvas of the screen. Students will use this knowledge to write programs to create 2D graphical patterns and colors on the screen.

Stage 1 – Desired Results

Established Goals:

New Jersey Student Learning Standards in Computer Design (2020)

8.1 Computer Science by the End of Grade 12

8.1.12.CS.1: Describe ways in which integrated systems hide underlying implementation details to simplify user experiences.
8.1.12.CS.2: Model interactions between application software, system software, and hardware.
8.1.12.CS.3: Compare the functions of application software, system software, and hardware.
8.1.12.CS.4: Develop guidelines that convey systematic troubleshooting strategies that others can use to identify and fix errors.
8.1.12.DA.2: Describe the trade-offs in how and where data is organized and stored.
8.1.12.DA.3: Translate between decimal numbers and binary numbers
8.1.12.DA.4: Explain the relationship between binary numbers and the storage and use of data in a computing device.
8.1.8.AP.1: Design and illustrate algorithms that solve complex problems using flowcharts and/or pseudocode.
8.1.12.AP.2: Create generalized computational solutions using collections instead of repeatedly using simple variables.
8.1.12.AP.3: Select and combine control structures for a specific application based upon performance and readability, and identify trade-offs to justify the choice.
8.1.12.AP.5: Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.
8.1.12.AP.6: Create artifacts by using procedures within a program, combinations of data and procedures, or independent but interrelated programs.
8.1.12.AP.8: Evaluate and refine computational artifacts to make them more usable and accessible.

New Jersey Student Learning Standards in Design Thinking (2020)

8.2 Design Thinking by the End of Grade 12

8.2.12.ED.5: Evaluate the effectiveness of a product or system based on factors that are related to its requirements, specifications, and constraints (e.g., safety, reliability, economic considerations, quality control, environmental concerns, manufacturability, maintenance and repair, ergonomics).
8.2.12.NT.1: Explain how different groups can contribute to the overall design of a product.

Essential Questions:

- What is java.awt and how is it important in graphics programming?
- What are pixels and how are they used in graphics programming? How is this representation related to concepts learned in mathematics?
- How are drawings created on the screen in Java?

Enduring Understandings:

Students will know…

- Java.awt is the Java abstract window toolkit which provides a package for drawing 2D graphics.
- A pixel is a graphical coordinate corresponding to a dot on the screen.
- Java graphics libraries provide classes that can be used in Java programs to create drawings on the screen.

Knowledge:

Students will know that…

- In Java, graphics are created by modifying attributes of graphical coordinates on the screen.
- Java provides a toolkit of methods which programmers can use to create color and drawings on the screen.

Skills:

Students will be able to…

- Represent a location on the screen with Java graphical coordinates.
- Utilize a Java graphics toolkit to draw lines and add color to write programs to create 2D drawings on the screen.

Interdisciplinary Connections

New Jersey Student Learning Standards for English Language Arts (2016)

NJSLSA.W8. Gather relevant information from multiple print and digital sources, assess the credibility and accuracy of each source, and integrate the information while avoiding plagiarism.

Standards for Mathematical Practices (2016)

1. Make sense of problems and persevere in solving them.
2. Reason abstractly and quantitatively.
3. Construct viable arguments and critique the reasoning of others.
4. Model with mathematics.
5. Use appropriate tools strategically.
6. Attend to precision.
7. Look for and make use of structure.
8. Look for and express regularity in repeated reasoning.

NJSLS Career Readiness,Life Literacies, and Key Skills (2020)

9.1.12.FP.5: Evaluate how behavioral bias (e.g., overconfidence, confirmation, recency, loss aversion, etc.) affects decision-making.

9.2.12.CAP.5: Assess and modify a personal plan to support current interests and postsecondary plans.

9.2.12.CAP.6: Identify transferable skills in career choices and design alternative career plans based on those skills.

9.4.12.CI.1: Demonstrate the ability to reflect, analyze, and use creative skills and ideas.

9.4.12.CI.2: Identify career pathways that highlight personal talents, skills, and abilities.

Critical Thinking and Problem-Solving

9.4.12.CT.1: Identify problem-solving strategies used in the development of an innovative product or practice (e.g., 1.1.12acc.C1b, 2.2.12.PF.3).

9.4.12.CT.2: Explain the potential benefits of collaborating to enhance critical thinking and problem solving)

9.4.12.TL.1: Assess digital tools based on features such as accessibility options, capacities, and utility for accomplishing a specified task.

Student Resources

Texts:

Think Java – How to Think Like a Computer Scientist Version 6.1.0 (2016 free text download) http://greenteapress.com/thinkjava6/thinkjava.pdf

Technology:

Java Integrated Development Environment (Eclipse or DrJava)

Websites:

Help with Java syntax and logic:
https://stackoverflow.com
https://geeksforgeeks.com
https://docs.oracle.com/en/java

Teacher Resources

Texts:

Think Java – How to Think Like a Computer Scientist Version 6.1.0 (2016 free text download)
http://greenteapress.com/thinkjava6/thinkjava.pdf

Technology:

Java language download.
Java Integrated Development Environment (Eclipse or DrJava)

Websites:

https://java/com
drjava.org
https://stackoverflow.com
https://geeksforgeeks.com
https://docs.oracle.com/en/java

Stage 2 – Assessment Evidence

Formative Assessments:

- Anticipatory and Do Now assignments
- Peer assessments
- Self-Assessment
- Closure assessments
- Informal Observations

Summative Assessments:

- Quizzes
- Java Programs

Performance Task(s): Moire Patterns

Moire patterns are patterns of lines that seem to shift around on the screen as you look at them from different angles. Students will research Moire patterns. Students will use 2D graphics to write a program to create Moire patterns on the screen.

Write java programs to

- Draw shapes of different colors on the screen.
- Create a Fractal Mickey/
- Use mouse events to change colors of objects on the screen.
- Create and etch-a-sketch program using the arrow keys for direction and function keys for color.
- Modify the Snake game parameters to personalize the game.

Stage 3 – Learning Plan

Week 1, 2: Graphics in Java
Lesson #1: Introduction to 2D graphics in Java

- Demonstrate how to represent Java graphical units (point, line, canvas) and translate them to the screen.
- Discuss and model how to create 2D graphics using Java library methods.
- Guide students to experiment with adding color to graphics.

Week 3: Drawings
Lesson #2: Creating Drawings in Java

- Illustrate methods for drawing a circle and other shapes on the screen.
- Discuss how to modify a drawing to create a fractal image.
- Discuss Moire pattern, which uses a series of lines and curves on the screen. Guide students to experiment with methods to create Moire patterns.

Weeks 4-5: Event Handlers, Modifying Attributes of a Game
Lesson #3: Mouse and Function Key Event Handlers

- Use the mouse event handler to cause changes based on mouse click, press, release, enter, exit events.
- Use the function key event handler to move an image on the screen and change colors.
- Modify the snake game to use a bigger board, move at a different speed, and use different pictures for images in the game.

Accommodations and Modifications:
Below please find a list of suggestions for accommodations and modifications to meet the diverse needs of our students. Teachers should consider this a resource and understand that they are not limited to the recommendations included below.

An accommodation changes HOW a student learns; the change needed does not alter the grade-level standard. A modification changes WHAT a student learns; the change alters the grade-level expectation.

***Special Education and 504 Plans**:*
All modifications and accommodations must be specific to each individual child's IEP (Individualized Educational Plan) or 504 Plan. All students with 504 plans should be provided the necessary tools to be successful in the course.

- Pre-teach or preview vocabulary
- Repeat or reword directions
- Have students repeat directions
- Use of small group instruction
- Pair visual prompts with verbal presentations
- Ask students to restate information, directions, and assignments
- Repetition and time for additional practice
- Model skills/techniques to be mastered
- Extended time to complete task/assignment/work
- Provide a copy of class notes Strategic seating (with a purpose – e.g. less distraction)
- Flexible seating
- Repetition and additional practice
- Use of manipulatives
- Use of assistive technology (as appropriate)
- Assign a peer buddy
- Emphasize key words or critical information by highlighting
- Use of graphic organizers
- Scaffold with prompts for sentence starters
- Check for understanding with more frequency
- Provide oral reminders and check student work during independent practice
- Chunk the assignment - broken up into smaller units, work submitted in phases
  Encourage student to proofread assignments and tests
- Provide regular home/school communication
- Teacher checks student planner
- Provide student with clear expectations in writing and grading criteria for assignments (rubrics)

Testing Accommodations:
Students should receive all testing accommodations for Benchmark assessments that they receive for State testing.

- Setting: Alternate setting for assessments, small groups, screens to block distractions
- Presentation: large print, test readers, use of audio, fewer questions on each page

- Response: answer verbally, use large block answer sheet, speech-to-text dictation, accept short answers
- Allow for retakes
- Provide study guides
- Use of reference aids such as glossary, multiplication tables, calculator
- Alternate ways to evaluate (projects or oral presentations instead of written tests)
- Open-book or open-note tests

English Language Learners:
All modifications and accommodations should be specific to each individual child's LEP level as determined by the WIDA screening or ACCESS, utilizing the WIDA Can Do Descriptors.

- Pre-teach or preview vocabulary
- Repeat or reword directions
- Have students repeat directions
- Use of small group instruction
- Scaffold language based on their Can Do Descriptors
- Alter materials and requirements according to Can Do Descriptors
- Adjust number of paragraphs or length of writing according to their Can Do Descriptor
- TPR (Total Physical Response-Sheltered Instruction strategy) Demonstrate concepts through multi-sensory forms such as with body language, intonation
- Pair visual prompts with verbal presentations
- Repetition and additional practice
- Model skills and techniques to be mastered
- Native Language translation (peer, assistive technology, bilingual dictionary)
- Emphasize key words or critical information by highlighting
- Use of graphic organizers
- Scaffold with prompts for sentence starters
- Check for understanding with more frequency
- Use of self-assessment rubrics Increase one-on-one conferencing; frequent check ins
- Use study guide to organize materials
- Make vocabulary words available in a student created vocabulary notebook, vocabulary bank, Word Wall, or vocabulary ring
- Extended time
- Select text complexity and tiered vocabulary according to Can Do Descriptors
- Projects completed individually or with partners
- Use an online dictionary that includes images for words.

Students at Risk of Failure:
- Use of self-assessment rubrics for check-in
- Pair visual prompts with verbal presentations
- Ask students to restate information and/or directions
- Opportunity for repetition and additional practice
- Model skills/techniques to be mastered

- Extended time
- Provide copy of class notes
- Strategic seating with a purpose
- Provide students opportunity to make corrections and/or explain their answers
- Support organizational skills

High Achieving:

- Allow for student choice from a menu of differentiated outcomes; choices grouped by complexity of thinking skills; variety of options enable students to work in the mode that most interests them
- Allow students to pursue independent projects based on their individual interests
- Provide enrichment activities that include more complex material
- Allow opportunities for peer collaboration and team-teaching
- Set individual goals
- Conduct research and provide presentation of appropriate topics

Pacing Guide

Day	Topic	Standard
1	What is programming/language/computer science	8.1.12.CS.1
2	Programming environment	8.1.12.CS.3
3	First program, displaying strings, escape sequences, formatting, debugging	8.1.12.CS.2
4	Quiz, Performance Task	8.1.12.CS.2
5	Variables, variable assignment, state diagrams, printing variables	8.1.12.CS.2
6	Keyword i/o, operators, operations, strings	8.1.12.CS.2
7	Composition of statements/expressions, types of errors	8.1.12.CS.4
8	Quiz/performance task	8.1.12.CS.3
9	Performance task	8.1.12.CS.3
10	System and scanner class, program structure	8.1.12.CS.2
11	Literals and constants, formatting output, modulus operator	8.1.12.CS.2
12	Putting it together, scanner bug	8.1.12.CS.2
13	Quiz/performance task	8.1.12.CS.3
14	Performance task	8.1.12.CS.3
15	Math methods, composition, new methods, flow of execution	8.1.12.AP.3
16	Parameters/arguments, stack diagrams	8.1.12.AP.3
17	Documentation	8.1.12.CS.2
18	Quiz/performance task	8.1.12.CS.2
19	Performance task	8.1.12.CS.2
20	Relational/Logical operators, conditionals, nesting	8.1.12.CS.2
21	Flags, return, validation	8.1.12.CS.2
22	Performance Task	8.1.12.CS.2

23	Conditionals, alternate execution	8.1.12.CS.2
24	Chained/nested conditionals, return statement	8.1.12.AP.3
25	Recursion	8.1.12.CS.2
26	Binary numbers	8.1.12.DA.3, 8.1.12.DA.4
27	Stack diagrams	8.1.12.CS.2
28	Performance task	8.1.12.CS.2
29	Quiz, performance task	8.1.12.CS.2
30	Performance task	8.1.12.CS.2
31	Return values	8.1.12.AP.3
32	Program development	8.1.12.CS.2
33	Modulus operator	8.1.12.CS.2
34	Composition	8.1.12.CS.2
35	Overloading	8.1.12.CS.2
36	Boolean operators and expressions	8.1.12.CS.2
37	Boolean expressions	8.1.12.CS.2
38	Flow of execution	8.1.12.CS.2
39	Quiz, performance task	8.1.12.CS.2
40	Performance task	8.1.12.CS.2
41	Review for midterm	8.1.12.CS.2
42	Review for midterm	8.1.12.CS.2
43	Midterm	8.1.12.CS.2
44	Midterm	8.1.12.CS.2
45	While statements, tables	8.1.12.CS.2
46	Encapsulation	8.1.12.DA.2, 8.1.8.AP.1, 8.1.12.AP.8

47	More generalization	8.1.12.DA.2, 8.1.8.AP.1, 8.1.12.AP.8
48	For-loop	8.1.12.AP.3
49	Do-while loop	8.1.12.AP.3
50	Break, continue	8.1.12.AP.3, 8.1.12.AP.8
51	Quiz, performance task	8.1.12.AP.3
52	Performance task	8.1.12.AP.3
53	Arrays	8.1.12.DA.6, 8.1.12.AP.2
54	Array traversal	8.1.12.AP.2
55	Tables and arrays	8.1.12.AP.2
56	Quiz, performance task	8.1.12.AP.2
57	Performance task	8.1.12.AP.2
58	Strings, string traversal, characters	8.1.12.CS.2
59	Substrings, index of a string	8.1.12.CS.2
60	String comparison	8.1.12.CS.2
61	Formatting strings	8.1.12.AP.3
62	Quiz, performance task	8.1.12.CS.2
63	Performance task	8.1.12.CS.2
64	Objects, attributes, objects as parameters/return types	8.1.12.CS.2
65	Attributes	8.1.12.CS.2
66	Objects as parameters/return types	8.1.12.CS.2
67	Mutable objects, aliasing	8.1.12.CS.2
68	Null keyword	8.1.12.CS.2
69	Quiz, performance task	8.1.12.CS.2

70	Performance task	8.1.12.AP.3
72	Classes	8.1.12.CS.2
73	Constructors	8.1.12.CS.2
74	Getters, setters	8.1.12.CS.2
75	Quiz, performance task	8.1.12.CS.2
76	Performance task	8.1.12.CS.2
77	Dialog box i/o	8.1.12.AP.3
78	Graphics Methods	8.1.12.AP.3
79	Drawing methods - lines, squares, rectangles	8.1.12.CS.2
80	Drawing methods - circles, ellipses	8.1.12.CS.2
81	Setting color	8.1.12.CS.2
82	Mouse events	8.1.12.CS.2
83	Keyboard events - arrow keys	8.1.12.CS.2
84	Keyboard events - Function keys	8.1.12.CS.2
85	Analyzing a small computer game	8.2.12.ED.5
86	Attributes of a small computer game	8.2.12.ED.5
87	Review for final	8.1.12.CS.2
88	Review for final	8.1.12.CS.2
89	Final	8.1.12.CS.2
90	Final	8.1.12.CS.2

NEPTUNE TOWNSHIP SCHOOL DISTRICT
Office of the Superintendent
60 Neptune Blvd.
Neptune, NJ 07753

An Affirmative Action Equal Opportunity Employer

2023