



MIDLAND PARK PUBLIC SCHOOLS
Midland Park, New Jersey
CURRICULUM

Computer Programming I

Prepared by:
Danielle Vandenberghe

Superintendent of Schools:
Marie C. Cirasella, Ed.D.
Director of Curriculum, Instruction & Assessment:
Melissa Quackenbush

*Approved by the Midland Park Board of Education on
August 16, 2016*

Computer Programming I

Course Description:

Computer Programming I is a beginning programming class using the Java programming language. Java provides an excellent environment for the beginning programmer as a student can quickly build useful programs while learning the basics of structured and object-oriented programming techniques. This class assumes that you have little or no programming experience. It provides a solid background in good object-oriented programming techniques and introduces terminology using clear, familiar language. Upon the completion of the class students will understand concepts used in object-oriented programming and be able to modify and create simple Java programs. Students will also possess a fundamental knowledge of object-oriented programming, which will serve them well in advanced Java courses or in studying other object-oriented languages.

Suggested Course Sequence:

Unit 1: Introduction to Computers, Programs, and Java - about 1 week

Unit 2: Creating Java Programs - about 4 ½ weeks

Unit 3: Using Data - about 5 weeks

Unit 4: Using Methods, Classes, and Objects - about 5 ½ weeks

Unit 5: More Object Concepts - about 5 weeks

Unit 6: Making Decisions - about 5 weeks

Unit 7: Looping - about 5 weeks

Unit 8: Characters, Strings, and StringBuilder - about 4 weeks

Unit 9: Arrays - about 4 ½ weeks

Prerequisite: Intro to Computers or Approval from Computer Teacher/Principal

Content Area: Computer Science	
Unit Title: Computer Programming I - Unit 1 - Introduction to Computers, Programs, and Java	
Grade Level: 10-12	
<p>Unit Summary: This unit provides an introduction to computers basics, programs, and operating systems. Students will learn some of the history of computers and how computers have evolved. Students will also learn about the relationship between Java and the World Wide Web. The unit also provides a tutorial on how to develop Java programs using NetBeans.</p> <p>Interdisciplinary Connections: Math: integers, decimals, Boolean numbers and other mathematical referenced and connections. English: connections to basic forms of speech (nouns, verbs, adjectives), vocabulary and research skills.</p> <p>21st Century Themes and Skills: Creativity and Innovation, Communication and Collaboration, Critical Thinking & Problem Solving, Information, Media, and Technology Skills, Life and Career Skills.</p> <p>CRP1. Act as a responsible and contributing citizen and employee.</p> <p>CRP2. Apply appropriate academic and technical skills.</p> <p>CRP4. Communicate clearly and effectively</p> <p>CRP6. Demonstrate creativity and innovation.</p> <p>CRP7. Employ valid and reliable research strategies.</p> <p>CRP8. Utilize critical thinking to make sense of problems and persevere in solving them.</p> <p>CRP11. Use technology to enhance productivity.</p>	
Standards (Content and Technology):	
CPI#:	Statement:
8.1.12.A.1	Create a personal digital portfolio which reflects personal and academic interests, achievements, and career aspirations by using a variety of digital tools and resources.
8.1.12.A.3	Collaborate in online courses, learning communities, social networks or virtual worlds to discuss a resolution to a problem or issue.
8.1.12.B.2	Apply previous content knowledge by creating and piloting a digital learning game or tutorial.
8.1.12.B.4	Investigate a technology used in a given period of history, e.g., stone age, industrial revolution or information age, and identify their impact and how they may have changed to meet human needs and wants.
8.1.12.F.1	Evaluate the strengths and limitations of emerging technologies and their impact on educational, career, personal and or social needs.
8.2.12.E.1	Demonstrate an understanding of the problem-solving capacity of computers in our world.
8.2.12.E.2	Analyze the relationships between internal and external computer components
8.2.12.E.3	Use a programming language to solve problems or accomplish a task (e.g., robotic functions, website designs, applications, and games
8.2.12.E.4	Use appropriate terms in conversation (e.g., troubleshooting, peripherals, diagnostic software, GUI, abstractions, variables, data types and conditional statements).
Unit Essential Question(s): <ul style="list-style-type: none"> What are the parts of a computer? How does a computer process information? How do you run a Java program using NetBeans? 	Unit Enduring Understandings: <ul style="list-style-type: none"> Students will learn the parts of a computer, how they interact with each other, and how a computer process information. Students will learn how to run a Java program using NetBeans.
Unit Learning Targets/Objectives: <i>Students will</i> <ul style="list-style-type: none"> Understand computer basics, programs, and operating systems. 	

- Understand how a computer processes information
- Describe the relationship between Java and the World Wide Web
- Learn the history of computers and programming
- Learn NetBeans for Java programming

Formative Assessments:

- Teacher observations
- Practice programs
- Discussions

Summative/Benchmark Assessment(s):

- Checkpoint questions/quizzes
- Research Paper
- Programs
- Current Event Journal

Resources/Materials (copy hyperlinks for digital resources):

- Text Book: *Introduction to Java Programming* by Y. Daniel Liang
http://wps.pearsoned.com/ecs_liang_iip_10/244/62489/15997433.cw/index.html
- <http://csunplugged.com/>
- www.codehs.com

Modifications:

Special Education Students

- Allow errors
- Rephrase questions, directions, and explanations
- Allow extended time to answer questions, and permit drawing, as an explanation
- Accept participation at any level, even one word
- Consult with Case Managers and follow IEP accommodations/modifications

At-Risk Students

- Provide extended time to complete tasks
- Consult with Guidance Counselors and follow I&RS procedures/action plans
- Consult with classroom teacher(s) for specific behavior interventions
- Provide rewards as necessary

English Language Learners

- Assign a buddy, same language or English speaking
- Allow errors in speaking
- Rephrase questions, directions, and explanations
- Allow extended time to answer questions
- Accept participation at any level, even one word

Gifted and Talented Students

- Provide extension activities
- Build on students' intrinsic motivations
- Consult with parents to accommodate students' interests in completing tasks at their level of engagement

Lesson Name/ Topic	Lesson Objective(s)	Time frame (day(s) to complete)
What is a computer	<ul style="list-style-type: none"> ● Learn what a computer is ● Learn what the central processing unit is ● Learn about bits and bytes ● Learn about a computer's memory ● Learn about different storage devices 	1 day

	<ul style="list-style-type: none"> ● Learn about input and output devices ● Learn about communication devices 	
Binary Numbers	<ul style="list-style-type: none"> ● Learn about binary numbers more in-depth 	1 day
History	<ul style="list-style-type: none"> ● Learn about the history of computers and programming languages ● Research paper 	1 day (time will be given for research paper each week)
Program- ming Languages	<ul style="list-style-type: none"> ● Learn about Machine Language and Assembly Language ● Learn about High-Level Language 	1 day
Operating Systems	<ul style="list-style-type: none"> ● Learn about a computer's operating system ● Learn about controlling and monitoring system activities ● Learn about allocating and assigning system resources ● Learn about scheduling operations 	½ day
Java, the World Wide Web, and Beyond	<ul style="list-style-type: none"> ● Learn about the relationship between Java and the World Wide Web 	½ day

Teacher Notes: Program time may vary depending on the complexity of the programs given.

Additional Resources

Click links below to access additional resources used to design this unit:

Text Book: *Java Programming by Joyce Farrell*

<https://login.cengage.com/cb/login.htm>

<https://code.org/>

<http://www.i-programmer.info/news.html>

<http://www.computerscienceonline.org/cs-programs-before-college/>

Content Area: Computer Science	
Unit Title: Computer Programming I - Unit 2 - Creating Java Programs	
Grade Level: 10-12	
<p>Unit Summary: This unit provides an introduction to programming. Students will learn basic programming terminology and apply this to the Java programming environment. They will also learn how to create a simple Java application that produces output to the console. This unit covers the basic components of a Java application and how to compile and run a Java program.</p> <p>Interdisciplinary Connections: Math: integers, decimals, Boolean numbers and other mathematical referenced and connections. English: connections to basic forms of speech (nouns, verbs, adjectives), and vocabulary.</p> <p>21st Century Themes and Skills: Creativity and Innovation, Communication and Collaboration, Critical Thinking & Problem Solving, Information, Media, and Technology Skills, Life and Career Skills.</p> <p>CRP1. Act as a responsible and contributing citizen and employee.</p> <p>CRP2. Apply appropriate academic and technical skills.</p> <p>CRP4. Communicate clearly and effectively</p> <p>CRP6. Demonstrate creativity and innovation.</p> <p>CRP8. Utilize critical thinking to make sense of problems and persevere in solving them.</p> <p>CRP11. Use technology to enhance productivity.</p>	
Standards (Content and Technology):	
CPI#:	Statement:
8.1.12.A.1	Create a personal digital portfolio which reflects personal and academic interests, achievements, and career aspirations by using a variety of digital tools and resources.
8.1.12.A.3	Collaborate in online courses, learning communities, social networks or virtual worlds to discuss a resolution to a problem or issue.
8.1.12.B.2	Apply previous content knowledge by creating and piloting a digital learning game or tutorial.
8.1.12.F.1	Evaluate the strengths and limitations of emerging technologies and their impact on educational, career, personal and or social needs.
8.2.12.E.1	Demonstrate an understanding of the problem-solving capacity of computers in our world.
8.2.12.E.3	Use a programming language to solve problems or accomplish a task (e.g., robotic functions, website designs, applications, and games
8.2.12.E.4	Use appropriate terms in conversation (e.g., troubleshooting, peripherals, diagnostic software, GUI, abstractions, variables, data types and conditional statements).
Unit Essential Question(s): <ul style="list-style-type: none"> What is basic programing terminology? How do you create a simple Java application? How do you compile and run a Java program? 	Unit Enduring Understandings: <ul style="list-style-type: none"> Students will learn basic programming terminology. Students will learn how to create, compile and run a simple Java application/program.
Unit Learning Targets/Objectives: <i>Students will</i> <ul style="list-style-type: none"> Define basic programming terminology Compare procedural and object-oriented programming Describe the features of the Java programming language Analyze a Java application that produces console output Compile a Java class and correct syntax errors Run a Java application and correct logic errors 	

- Add comments to a Java class
- Create a Java application that produces GUI output
- Find help

Formative Assessments:

- Teacher observations
- Practice programs
- Discussions

Summative/Benchmark Assessment(s):

- Checkpoint questions/quizzes
- Unit tests
- Programming Projects
- Chapter Portfolio
- Current Event Journal

Resources/Materials (copy hyperlinks for digital resources):

- Text Book: *Java Programming by Joyce Farrell*
- <https://login.cengage.com/cb/login.htm>
- www.codehs.com

Modifications:

Special Education Students

- Allow errors
- Rephrase questions, directions, and explanations
- Allow extended time to answer questions, and permit drawing, as an explanation
- Accept participation at any level, even one word
- Consult with Case Managers and follow IEP accommodations/modifications

At-Risk Students

- Provide extended time to complete tasks
- Consult with Guidance Counselors and follow I&RS procedures/action plans
- Consult with classroom teacher(s) for specific behavior interventions
- Provide rewards as necessary

English Language Learners

- Assign a buddy, same language or English speaking
- Allow errors in speaking
- Rephrase questions, directions, and explanations
- Allow extended time to answer questions
- Accept participation at any level, even one word

Gifted and Talented Students

- Provide extension activities
- Build on students' intrinsic motivations
- Consult with parents to accommodate students' interests in completing tasks at their level of engagement

Lesson Name/ Topic	Lesson Objective(s)	Time frame (day(s) to complete)
Learning Program- ming Termin- ology	● Learn important programming terminology and how a program is executed	1 day
Comparing Procedural and Object-	● Learn about and compare procedural and object-oriented programming	1 day

Oriented Program- ming Concepts	<ul style="list-style-type: none"> ● Understand classes, objects, and encapsulation ● Understand Inheritance and Polymorphism 	
Features of the Java Program- ming Language	<ul style="list-style-type: none"> ● Learn about the features of the Java programming language and what to use to run a program 	1 day
Analyzing a Java Application that Produces Console Output	<ul style="list-style-type: none"> ● Learn how to read a java application, identify parts, and how to write an application ● Understand what First Class is ● Understanding Indentation ● Understand the main() Method ● How to save a Java class 	2 day
Compiling a Java class and Correcting Syntax Errors	<ul style="list-style-type: none"> ● Learn how to compile a java class and correct syntax errors 	2 days
Running a Java Application and Correcting Logic Errors	<ul style="list-style-type: none"> ● Learn how to run a java application and correct logic errors ● Learn how to modify a compiled Java class 	1 day
Adding Comments to a Java Class	<ul style="list-style-type: none"> ● Learn how to add comments to a java class 	1 day
Creating a Java Application that Produces GUI Output	<ul style="list-style-type: none"> ● Learn how to create a java application that produces GUI output 	1 day
Finding Help/Don't Do it	<ul style="list-style-type: none"> ● Find where to get helpful information ● Look at common mistakes and errors 	1 day
Chapter Portfolio	<ul style="list-style-type: none"> ● Create a chapter portfolio containing vocabulary, reviews, and programming/debugging exercises. 	5 days
Programs	<ul style="list-style-type: none"> ● Game Program ● Case Study 	5 days

Teacher Notes: Program time may vary depending on the complexity of the programs given.

Additional Resources

Click links below to access additional resources used to design this unit:

<http://csunplugged.com/>

<https://code.org/>

<http://www.i-programmer.info/news.html>

<http://www.computerscienceonline.org/cs-programs-before-college/>

Content Area: Computer Science	
Unit Title: Computer Programming I - Unit 3 - Using Data	
Grade Level: 10-12	
<p>Unit Summary: Students will learn the eight primitive data types in the Java language. Students will learn to work with integer, floating-point, Boolean, and character values. Arithmetic and comparison operators are introduced. Finally, students will learn to create input and confirm dialog boxes using the JOptionPane class.</p> <p>Interdisciplinary Connections: Math: integers, decimals, Boolean numbers and other mathematical referenced and connections. English: connections to basic forms of speech (nouns, verbs, adjectives), and vocabulary.</p> <p>21st Century Themes and Skills: Creativity and Innovation, Communication and Collaboration, Critical Thinking & Problem Solving, Information, Media, and Technology Skills, Life and Career Skills.</p> <p>CRP1. Act as a responsible and contributing citizen and employee.</p> <p>CRP2. Apply appropriate academic and technical skills.</p> <p>CRP4. Communicate clearly and effectively</p> <p>CRP6. Demonstrate creativity and innovation.</p> <p>CRP8. Utilize critical thinking to make sense of problems and persevere in solving them.</p> <p>CRP11. Use technology to enhance productivity.</p>	
Standards (Content and Technology):	
CPI#:	Statement:
8.1.12.A.1	Create a personal digital portfolio which reflects personal and academic interests, achievements, and career aspirations by using a variety of digital tools and resources.
8.1.12.A.3	Collaborate in online courses, learning communities, social networks or virtual worlds to discuss a resolution to a problem or issue.
8.1.12.B.2	Apply previous content knowledge by creating and piloting a digital learning game or tutorial.
8.1.12.F.1	Evaluate the strengths and limitations of emerging technologies and their impact on educational, career, personal and or social needs.
8.2.12.E.1	Demonstrate an understanding of the problem-solving capacity of computers in our world.
8.2.12.E.3	Use a programming language to solve problems or accomplish a task (e.g., robotic functions, website designs, applications, and games
8.2.12.E.4	Use appropriate terms in conversation (e.g., troubleshooting, peripherals, diagnostic software, GUI, abstractions, variables, data types and conditional statements).
Unit Essential Question(s): <ul style="list-style-type: none"> What are the different data types that can be used? How is user input accepted by the user? How can you perform arithmetic and type conversions? 	Unit Enduring Understandings: <ul style="list-style-type: none"> Students will learn the four different data types. Students will learn how to input user data into a program. Students will learn how to perform arithmetic and convert between data types.
Unit Learning Targets/Objectives: <i>Students will</i> <ul style="list-style-type: none"> Declare and use constants and variables Use integer data types Use the boolean data type Use floating-point data types Use the char data type Use the Scanner class to accept keyboard input 	

- Use the JOptionPane class to accept GUI input
- Perform arithmetic
- Understand type conversion

Formative Assessments:

- Teacher observations
- Practice programs
- Discussions

Summative/Benchmark Assessment(s):

- Checkpoint questions/quizzes
- Unit tests
- Programming Projects
- Chapter Portfolio
- Current Event Journal

Resources/Materials (copy hyperlinks for digital resources):

- Text Book: *Java Programming by Joyce Farrell*
- <https://login.cengage.com/cb/login.htm>
- www.codehs.com

Modifications:
Special Education Students

- Allow errors
- Rephrase questions, directions, and explanations
- Allow extended time to answer questions, and permit drawing, as an explanation
- Accept participation at any level, even one word
- Consult with Case Managers and follow IEP accommodations/modifications

At-Risk Students

- Provide extended time to complete tasks
- Consult with Guidance Counselors and follow I&RS procedures/action plans
- Consult with classroom teacher(s) for specific behavior interventions
- Provide rewards as necessary

English Language Learners

- Assign a buddy, same language or English speaking
- Allow errors in speaking
- Rephrase questions, directions, and explanations
- Allow extended time to answer questions
- Accept participation at any level, even one word

Gifted and Talented Students

- Provide extension activities
- Build on students' intrinsic motivations
- Consult with parents to accommodate students' interests in completing tasks at their level of engagement

Lesson Name/Topic	Lesson Objective(s)	Time frame (day(s) to complete)
Declaring and Using Constants and Variables	<ul style="list-style-type: none"> ● Learn how to declare and use constants and variables in programs ● Learn how to combine strings to variables and constants 	2-3 days
Learning about Integer Data Types	<ul style="list-style-type: none"> ● Learn the different variations of the integer data type and when/how to use each one. 	1 day

Using the boolean Data Type	<ul style="list-style-type: none"> ● Learn how and when to use the boolean Data Type 	½ day
Learning About Floating-Point Data Types	<ul style="list-style-type: none"> ● Learn how and when to use the Floating-Point Data Type 	½ day
Using the char Data Type	<ul style="list-style-type: none"> ● Learn how and when to use the char Data Type 	1 day
Using the Scanner Class to Accept Keyboard Input	<ul style="list-style-type: none"> ● Learn how to use the scanner class to accept user input into programs ● Using the nextLine() with Scanner input methods. 	2 day
Using the JOptionPane Class to accept GUI Input	<ul style="list-style-type: none"> ● Learn how to use Input and Confirm dialog boxes 	1 day
Performing Arithmetic	<ul style="list-style-type: none"> ● Learn the five standard arithmetic operators. ● Understand associativity and precedence in arithmetic operators ● Writing arithmetic statements efficiently ● Understanding imprecision in floating-point numbers 	2 days
Understanding Type Conversion	<ul style="list-style-type: none"> ● Learn how to convert from one data type to another. ● Learn the difference between automatic type conversion and explicit type conversion. 	2 days
Chapter Portfolio	<ul style="list-style-type: none"> ● Create a chapter portfolio containing vocabulary, reviews, and programming/debugging exercises. 	5 days
Programs	<ul style="list-style-type: none"> ● Game Program ● Case Study 	5 days

Teacher Notes: Program time may vary depending on the complexity of the programs given.

Additional Resources

Click links below to access additional resources used to design this unit:

<http://csunplugged.com/>

<https://code.org/>

<http://www.i-programmer.info/news.html>

<http://www.computerscienceonline.org/cs-programs-before-college/>

Content Area: Computer Science	
Unit Title: Computer Programming I - Unit 4 - Using Methods, Classes, and Objects	
Grade Level: 10-12	
Unit Summary: This unit introduces students to the creation of classes, variables, and methods. Students will learn to create methods that accept arguments and return values. They will learn to create a class composed of instance variables and methods. Students will create and invoke constructor methods to initialize instances of a class.	
Interdisciplinary Connections: Math: integers, decimals, Boolean numbers and other mathematical referenced and connections. English: connections to basic forms of speech (nouns, verbs, adjectives), and vocabulary.	
21st Century Themes and Skills: Creativity and Innovation, Communication and Collaboration, Critical Thinking & Problem Solving, Information, Media, and Technology Skills, Life and Career Skills. CRP1. Act as a responsible and contributing citizen and employee. CRP2. Apply appropriate academic and technical skills. CRP4. Communicate clearly and effectively CRP6. Demonstrate creativity and innovation. CRP8. Utilize critical thinking to make sense of problems and persevere in solving them. CRP11. Use technology to enhance productivity.	
Standards (Content and Technology):	
CPI#:	Statement:
8.1.12.A.1	Create a personal digital portfolio which reflects personal and academic interests, achievements, and career aspirations by using a variety of digital tools and resources.
8.1.12.A.3	Collaborate in online courses, learning communities, social networks or virtual worlds to discuss a resolution to a problem or issue.
8.1.12.B.2	Apply previous content knowledge by creating and piloting a digital learning game or tutorial.
8.1.12.F.1	Evaluate the strengths and limitations of emerging technologies and their impact on educational, career, personal and or social needs.
8.2.12.E.1	Demonstrate an understanding of the problem-solving capacity of computers in our world.
8.2.12.E.3	Use a programming language to solve problems or accomplish a task (e.g., robotic functions, website designs, applications, and games)
8.2.12.E.4	Use appropriate terms in conversation (e.g., troubleshooting, peripherals, diagnostic software, GUI, abstractions, variables, data types and conditional statements).
Unit Essential Question(s): <ul style="list-style-type: none"> How do you create classes, variables and methods? How do method accept arguments and return values? 	Unit Enduring Understandings: <ul style="list-style-type: none"> Students will understand how to create classes, variables and methods. Students will be able to create a method that accepts arguments and returns values.
Unit Learning Targets/Objectives: <i>Students will</i> <ul style="list-style-type: none"> Learn about method calls and placement Identify the parts of a method Add parameters to methods Create methods that return values Learn about classes and objects Create a class Create instance methods in a class 	

- Declare objects and use their methods
- Create constructors
- Appreciate classes as data types

Formative Assessments:

- Teacher observations
- Practice programs
- Discussions

Summative/Benchmark Assessment(s):

- Checkpoint questions/quizzes
- Unit tests
- Programming Projects
- Chapter Portfolio
- Current Event Journal

Resources/Materials (copy hyperlinks for digital resources):

- Text Book: *Java Programming by Joyce Farrell*
- <https://login.cengage.com/cb/login.htm>
- www.codehs.com

Modifications:
Special Education Students

- Allow errors
- Rephrase questions, directions, and explanations
- Allow extended time to answer questions, and permit drawing, as an explanation
- Accept participation at any level, even one word
- Consult with Case Managers and follow IEP accommodations/modifications

At-Risk Students

- Provide extended time to complete tasks
- Consult with Guidance Counselors and follow I&RS procedures/action plans
- Consult with classroom teacher(s) for specific behavior interventions
- Provide rewards as necessary

English Language Learners

- Assign a buddy, same language or English speaking
- Allow errors in speaking
- Rephrase questions, directions, and explanations
- Allow extended time to answer questions
- Accept participation at any level, even one word

Gifted and Talented Students

- Provide extension activities
- Build on students' intrinsic motivations
- Consult with parents to accommodate students' interests in completing tasks at their level of engagement

Lesson Name/Topic	Lesson Objective(s)	Time frame (day(s) to complete)
Under-standing Method Calls and Placement	<ul style="list-style-type: none"> ● Learn what a method is and how to use it. 	1 day
Under-standing Method Con-	<ul style="list-style-type: none"> ● Learn how to construct a method within a program ● Learn how to access specifiers for a specific method. 	2 days

struction	<ul style="list-style-type: none"> ● Learn how to return a value ● Learn how to name a method ● Learn how to use parentheses within a method 	
Adding Parameters to Methods	<ul style="list-style-type: none"> ● Learn how to use arguments and parameters in programs ● Learn how to create a method that receives a single parameter ● Learn how to create a method that requires multiple parameters 	1 - 2 days
Creating Methods that Return Values	<ul style="list-style-type: none"> ● Learn how to return a value using a method ● Learn how to use methods within a method 	2 days
Learning about Classes and Objects	<ul style="list-style-type: none"> ● Understand the relationship between classes and objects. 	1 day
Creating a Class	<ul style="list-style-type: none"> ● Learn the proper way to create a class 	½ day
Creating Instance Methods in a Class	<ul style="list-style-type: none"> ● Learn how to create a method within a created class. ● Learn how to organize classes 	2 ½ day
Declaring Objects and Using their Methods	<ul style="list-style-type: none"> ● Learn how to declare an object within a class and use their methods. ● Understand data hiding 	2 day
An Introduction to Using Constructors	<ul style="list-style-type: none"> ● Learn what a constructor is and how it is used. 	1 day
Understanding the Classes are Data Types	<ul style="list-style-type: none"> ● Understand that created classes become data types 	2 days
Chapter Portfolio	<ul style="list-style-type: none"> ● Create a chapter portfolio containing vocabulary, reviews, and programming/debugging exercises. 	5 days
Programs	<ul style="list-style-type: none"> ● Game Program ● Case Study 	5 days

Teacher Notes: Program time may vary depending on the complexity of the programs given.

Additional Resources

Click links below to access additional resources used to design this unit:

<http://csunplugged.com/>

<https://code.org/>

<http://www.i-programmer.info/news.html>

<http://www.computerscienceonline.org/cs-programs-before-college/>

Content Area: Computer Science	
Unit Title: Computer Programming I - Unit 5 - More Object Concepts	
Grade Level: 10-12	
<p>Unit Summary: Students will be introduced to block and scope, overriding, and overloading. Students will learn to create overloaded methods and constructors. They will learn about static variables and how to create constants using the final keyword. Finally, students will learn to use prewritten classes in the java.lang and java.util packages, and other packages.</p> <p>Interdisciplinary Connections: Math: integers, decimals, Boolean numbers and other mathematical referenced and connections. English: connections to basic forms of speech (nouns, verbs, adjectives), and vocabulary.</p> <p>21st Century Themes and Skills: Creativity and Innovation, Communication and Collaboration, Critical Thinking & Problem Solving, Information, Media, and Technology Skills, Life and Career Skills.</p> <p>CRP1. Act as a responsible and contributing citizen and employee.</p> <p>CRP2. Apply appropriate academic and technical skills.</p> <p>CRP4. Communicate clearly and effectively</p> <p>CRP6. Demonstrate creativity and innovation.</p> <p>CRP8. Utilize critical thinking to make sense of problems and persevere in solving them.</p> <p>CRP11. Use technology to enhance productivity.</p>	
Standards (Content and Technology):	
CPI#:	Statement:
8.1.12.A.1	Create a personal digital portfolio which reflects personal and academic interests, achievements, and career aspirations by using a variety of digital tools and resources.
8.1.12.A.3	Collaborate in online courses, learning communities, social networks or virtual worlds to discuss a resolution to a problem or issue.
8.1.12.B.2	Apply previous content knowledge by creating and piloting a digital learning game or tutorial.
8.1.12.F.1	Evaluate the strengths and limitations of emerging technologies and their impact on educational, career, personal and or social needs.
8.2.12.E.1	Demonstrate an understanding of the problem-solving capacity of computers in our world.
8.2.12.E.3	Use a programming language to solve problems or accomplish a task (e.g., robotic functions, website designs, applications, and games
8.2.12.E.4	Use appropriate terms in conversation (e.g., troubleshooting, peripherals, diagnostic software, GUI, abstractions, variables, data types and conditional statements).
Unit Essential Question(s): <ul style="list-style-type: none"> How do we create overloaded methods and constructors? What are static variables? How do we use prewritten classes? 	Unit Enduring Understandings: <ul style="list-style-type: none"> Students will learn how to create and use overloaded methods and constructors. Students will learn how to use static variables. Students will learn how to use prewritten classes in their programs.
Unit Learning Targets/Objectives: <i>Students will</i> <ul style="list-style-type: none"> Understand blocks and scope Overload a method Avoid ambiguity Create and call constructors with parameters Use the this reference 	

- Use static fields
- Use automatically imported, prewritten constants and methods
- Use composition and nexts classes

Formative Assessments:

- Teacher observations
- Practice programs
- Discussions

Summative/Benchmark Assessment(s):

- Checkpoint questions/quizzes
- Unit tests
- Programming Projects
- Chapter Portfolio
- Current Event Journal

Resources/Materials (copy hyperlinks for digital resources):

- Text Book: *Java Programming by Joyce Farrell*
- <https://login.cengage.com/cb/login.htm>
- www.codehs.com

Modifications:
Special Education Students

- Allow errors
- Rephrase questions, directions, and explanations
- Allow extended time to answer questions, and permit drawing, as an explanation
- Accept participation at any level, even one word
- Consult with Case Managers and follow IEP accommodations/modifications

At-Risk Students

- Provide extended time to complete tasks
- Consult with Guidance Counselors and follow I&RS procedures/action plans
- Consult with classroom teacher(s) for specific behavior interventions
- Provide rewards as necessary

English Language Learners

- Assign a buddy, same language or English speaking
- Allow errors in speaking
- Rephrase questions, directions, and explanations
- Allow extended time to answer questions
- Accept participation at any level, even one word

Gifted and Talented Students

- Provide extension activities
- Build on students' intrinsic motivations
- Consult with parents to accommodate students' interests in completing tasks at their level of engagement

Lesson Name/ Topic	Lesson Objective(s)	Time frame (day(s) to complete)
Under-standing Blocks and Scope	<ul style="list-style-type: none"> ● Learn how to use blocks and scope in programs 	2 days
Over-loading a Method	<ul style="list-style-type: none"> ● Learn how to overload a method to use one identifier to execute diverse tasks ● Learn how to use automatic type promotion in method calls. 	2 days

Learning about Ambiguity	<ul style="list-style-type: none"> ● Understand what an ambiguous situation is and how to keep it from happening 	½ day
Creating and Calling Constructors with Parameters	<ul style="list-style-type: none"> ● Learn how to create and call constructors with parameters ● Learn about overloading constructors 	1 ½ day
Learning About the this Reference	<ul style="list-style-type: none"> ● Learn how to use the this reference ● Learn how to use the this reference to make overloaded constructors more efficient. 	3 days
Using static Fields	<ul style="list-style-type: none"> ● Learn how to use static fields in programing ● Learn how to use constant fields 	2 days
Using Auto-matically Imported, Prewritten Constants and Methods	<ul style="list-style-type: none"> ● Learn how to use prewritten constants and methods ● Learn about the math class ● Learn about importing classes that are not imported automatically ● Learn how to use the LocalDate class 	3 days
Understanding Composition and Nested Classes	<ul style="list-style-type: none"> ● Understand how to use composition to group classes ● Understand how to use nesting to group classes 	1 day
Chapter Portfolio	<ul style="list-style-type: none"> ● Create a chapter portfolio containing vocabulary, reviews, and programming/debugging exercises. 	5 days
Programs	<ul style="list-style-type: none"> ● Game Program ● Case Study 	5 days

Teacher Notes: Program time may vary depending on the complexity of the programs given.

Additional Resources

Click links below to access additional resources used to design this unit:

<http://csunplugged.com/>

<https://code.org/>

<http://www.i-programmer.info/news.html>

<http://www.computerscienceonline.org/cs-programs-before-college/>

ENGAGING STUDENTS • FOSTERING ACHIEVEMENT • CULTIVATING 21ST CENTURY GLOBAL SKILLS

Content Area: Computer Science	
Unit Title: Computer Programming I - Unit 6 - Making Decisions	
Grade Level: 10-12	
<p>Unit Summary: This unit introduces decision structures using the if, if ... else, and switch statements. Students will learn to execute program statements based on the result of a Boolean expression. They will also learn to use the logical operators AND, OR, and NOT, as well as the conditional operator. This unit presents a number of tips on avoiding common programming errors when making decisions.</p> <p>Interdisciplinary Connections: Math: integers, decimals, Boolean numbers and other mathematical referenced and connections. English: connections to basic forms of speech (nouns, verbs, adjectives), and vocabulary.</p> <p>21st Century Themes and Skills: Creativity and Innovation, Communication and Collaboration, Critical Thinking & Problem Solving, Information, Media, and Technology Skills, Life and Career Skills.</p> <p>CRP1. Act as a responsible and contributing citizen and employee.</p> <p>CRP2. Apply appropriate academic and technical skills.</p> <p>CRP4. Communicate clearly and effectively</p> <p>CRP6. Demonstrate creativity and innovation.</p> <p>CRP8. Utilize critical thinking to make sense of problems and persevere in solving them.</p> <p>CRP11. Use technology to enhance productivity.</p>	
Standards (Content and Technology):	
CPI#:	Statement:
8.1.12.A.1	Create a personal digital portfolio which reflects personal and academic interests, achievements, and career aspirations by using a variety of digital tools and resources.
8.1.12.A.3	Collaborate in online courses, learning communities, social networks or virtual worlds to discuss a resolution to a problem or issue.
8.1.12.B.2	Apply previous content knowledge by creating and piloting a digital learning game or tutorial.
8.1.12.F.1	Evaluate the strengths and limitations of emerging technologies and their impact on educational, career, personal and or social needs.
8.2.12.E.1	Demonstrate an understanding of the problem-solving capacity of computers in our world.
8.2.12.E.3	Use a programming language to solve problems or accomplish a task (e.g., robotic functions, website designs, applications, and games
8.2.12.E.4	Use appropriate terms in conversation (e.g., troubleshooting, peripherals, diagnostic software, GUI, abstractions, variables, data types and conditional statements).
Unit Essential Question(s): <ul style="list-style-type: none"> How are the if and if ... else statements used? How is the switch statement used? How are logic operators used? 	Unit Enduring Understandings: <ul style="list-style-type: none"> Students will learn to execute program statements based on the result of Boolean expression. Students will learn how to use the if and if ... else statements. Students will learn how to use the switch statement and logic operators.
Unit Learning Targets/Objectives: <i>Students will</i> <ul style="list-style-type: none"> Plan decision-making logic Make decisions with the if and if ... else statements Use multiple statements in if and if ... else clauses Nest if and if ... else statements 	

- Use AND and OR operators
- Make accurate and efficient decisions
- Use the switch statement
- Use the conditional and NOT operators
- Assess operator precedence
- Add decisions and constructors to instance methods

Formative Assessments:

- Teacher observations
- Practice programs
- Discussions

Summative/Benchmark Assessment(s):

- Checkpoint questions/quizzes
- Unit tests
- Programming Projects
- Chapter Portfolio
- Current Event Journal

Resources/Materials (copy hyperlinks for digital resources):

- Text Book: *Java Programming by Joyce Farrell*
- <https://login.cengage.com/cb/login.htm>
- www.codehs.com

Modifications:
Special Education Students

- Allow errors
- Rephrase questions, directions, and explanations
- Allow extended time to answer questions, and permit drawing, as an explanation
- Accept participation at any level, even one word
- Consult with Case Managers and follow IEP accommodations/modifications

At-Risk Students

- Provide extended time to complete tasks
- Consult with Guidance Counselors and follow I&RS procedures/action plans
- Consult with classroom teacher(s) for specific behavior interventions
- Provide rewards as necessary

English Language Learners

- Assign a buddy, same language or English speaking
- Allow errors in speaking
- Rephrase questions, directions, and explanations
- Allow extended time to answer questions
- Accept participation at any level, even one word

Gifted and Talented Students

- Provide extension activities
- Build on students' intrinsic motivations
- Consult with parents to accommodate students' interests in completing tasks at their level of engagement

Lesson Name/ Topic	Lesson Objective(s)	Time frame (day(s) to complete)
Planning Decision-Making Logic	<ul style="list-style-type: none"> ● Learn how to plan out a computer program 	1 day

The if and if ... else Statements	<ul style="list-style-type: none"> ● Learn how to use the if statement ● What to do with misplaced semicolons ● Using the assignment operator instead of the equivalency operator ● Comparing objects using the relational operators ● Learn about the if ... else statement 	2 days
Using Multiple Statements in if and if ... else Clauses	<ul style="list-style-type: none"> ● Learn how to use multiple statements in an if and if ... else statements 	2 days
Nesting if and if ... else Statements	<ul style="list-style-type: none"> ● Learn how to nest if and if ... else statements 	1 day
Using Logical AND and OR Operators	<ul style="list-style-type: none"> ● Learn how to use the AND operator ● Learn how to use the OR operator ● Learn what short-circuit evaluation is 	2 days
Making Accurate and Efficient Decisions	<ul style="list-style-type: none"> ● Learn how to make accurate range checks ● Learn how to make efficient range checks ● Learn how to use AND and the OR operators appropriately 	1 day
Using the switch Statement	<ul style="list-style-type: none"> ● Learn how to use the switch statement 	2 days
Using the Conditional and NOT Operators	<ul style="list-style-type: none"> ● Learn how to use the conditional operator ● Learn how to use the NOT operator 	½ day
Under-standing Operator Precedence	<ul style="list-style-type: none"> ● Understand operator precedence 	½ day
Adding Decisions and Constructors to Instance Methods	<ul style="list-style-type: none"> ● Learn how to add decisions and constructors to Instance methods 	2 days
Chapter Portfolio	<ul style="list-style-type: none"> ● Create a chapter portfolio containing vocabulary, reviews, 	5 days

	and programming/debugging exercises.	
Programs	<ul style="list-style-type: none"> ● Game Program ● Case Study 	5 days
<p>Teacher Notes: Program time may vary depending on the complexity of the programs given.</p> <p>Additional Resources Click links below to access additional resources used to design this unit: http://csunplugged.com/ https://code.org/ http://www.i-programmer.info/news.html http://www.computerscienceonline.org/cs-programs-before-college/</p>		

Content Area: Computer Science	
Unit Title: Computer Programming I - Unit 7 - Looping	
Grade Level: 10-12	
<p>Unit Summary: This unit covers looping structures. Students will learn to create definite and indefinite loops using the while statement. Next, they will learn to use Java's accumulating and incrementing operators. Students will use for loops to create a definite loop and do ... while loops for use when a posttest loop is required. Finally, students will learn how to create nested loops and how to improve loop efficiency.</p> <p>Interdisciplinary Connections: Math: integers, decimals, Boolean numbers and other mathematical referenced and connections. English: connections to basic forms of speech (nouns, verbs, adjectives), and vocabulary.</p> <p>21st Century Themes and Skills: Creativity and Innovation, Communication and Collaboration, Critical Thinking & Problem Solving, Information, Media, and Technology Skills, Life and Career Skills.</p> <p>CRP1. Act as a responsible and contributing citizen and employee.</p> <p>CRP2. Apply appropriate academic and technical skills.</p> <p>CRP4. Communicate clearly and effectively</p> <p>CRP6. Demonstrate creativity and innovation.</p> <p>CRP8. Utilize critical thinking to make sense of problems and persevere in solving them.</p> <p>CRP11. Use technology to enhance productivity.</p>	
Standards (Content and Technology):	
CPI#:	Statement:
8.1.12.A.1	Create a personal digital portfolio which reflects personal and academic interests, achievements, and career aspirations by using a variety of digital tools and resources.
8.1.12.A.3	Collaborate in online courses, learning communities, social networks or virtual worlds to discuss a resolution to a problem or issue.
8.1.12.B.2	Apply previous content knowledge by creating and piloting a digital learning game or tutorial.
8.1.12.F.1	Evaluate the strengths and limitations of emerging technologies and their impact on educational, career, personal and or social needs.
8.2.12.E.1	Demonstrate an understanding of the problem-solving capacity of computers in our world.
8.2.12.E.3	Use a programming language to solve problems or accomplish a task (e.g., robotic functions, website designs, applications, and games
8.2.12.E.4	Use appropriate terms in conversation (e.g., troubleshooting, peripherals, diagnostic software, GUI, abstractions, variables, data types and conditional statements).
Unit Essential Question(s): <ul style="list-style-type: none"> What is a looping structure? How do you use while, for and do ... while loops? How do you use loops efficiently? 	Unit Enduring Understandings: <ul style="list-style-type: none"> Students will learn how to create loops using while, for and do ... while statements. Students will learn how to improve loop efficiency.
Unit Learning Targets/Objectives: <i>Students will</i> <ul style="list-style-type: none"> Learn about the loop structure Create while loops Use shortcut arithmetic operators Create for loops Create do ... while loops Next loops Improve loop performance 	

Formative Assessments:

- Teacher observations
- Practice programs
- Discussions

Summative/Benchmark Assessment(s):

- Checkpoint questions/quizzes
- Unit tests
- Programming Projects
- Chapter Portfolio
- Current Event Journal

Resources/Materials (copy hyperlinks for digital resources):

- Text Book: *Java Programming* by Joyce Farrell
- <https://login.cengage.com/cb/login.htm>
- www.codehs.com

Modifications:
Special Education Students

- Allow errors
- Rephrase questions, directions, and explanations
- Allow extended time to answer questions, and permit drawing, as an explanation
- Accept participation at any level, even one word
- Consult with Case Managers and follow IEP accommodations/modifications

At-Risk Students

- Provide extended time to complete tasks
- Consult with Guidance Counselors and follow I&RS procedures/action plans
- Consult with classroom teacher(s) for specific behavior interventions
- Provide rewards as necessary

English Language Learners

- Assign a buddy, same language or English speaking
- Allow errors in speaking
- Rephrase questions, directions, and explanations
- Allow extended time to answer questions
- Accept participation at any level, even one word

Gifted and Talented Students

- Provide extension activities
- Build on students' intrinsic motivations
- Consult with parents to accommodate students' interests in completing tasks at their level of engagement

Lesson Name/ Topic	Lesson Objective(s)	Time frame (day(s) to complete)
Learning About the Loop Structure	<ul style="list-style-type: none"> ● Learn what a loop structure is 	1 day
Creating while Loops	<ul style="list-style-type: none"> ● Learn what a while loop is ● Learn how to write a definite while loop ● Learn how to avoid pitfalls of loops ● Learn how to alter a definite loop's control variable 	4 days

	<ul style="list-style-type: none"> ● Learn how to write an indefinite while loop ● Learn how to validate data 	
Using Shortcut Arithmetic Operators	<ul style="list-style-type: none"> ● Understand how to use shortcuts for arithmetic operators 	2 days
Creating a for Loop	<ul style="list-style-type: none"> ● Learn what a for loop is ● Understand unconventional for loops 	2 days
Learning How and When to Use a do ... while Loop	<ul style="list-style-type: none"> ● Learn what a do ... while loop is, how to use it and when to use it 	1 day
Learning about Nested Loops	<ul style="list-style-type: none"> ● Learn how to nest loops 	2 days
Improving Loop Performance	<ul style="list-style-type: none"> ● Understand how to improve loop performance ● Learn how to avoid unnecessary operations ● How to consider the order of evaluation of short-circuit operators ● How to compare to zero ● Learn how to employ loop fusion ● Understand how to use prefix incrementing rather than postfix 	3 days
Chapter Portfolio	<ul style="list-style-type: none"> ● Create a chapter portfolio containing vocabulary, reviews, and programming/debugging exercises. 	5 days
Programs	<ul style="list-style-type: none"> ● Game Program ● Case Study 	5 days

Teacher Notes: Program time may vary depending on the complexity of the programs given.

Additional Resources

Click links below to access additional resources used to design this unit:

<http://csunplugged.com/>

<https://code.org/>

<http://www.i-programmer.info/news.html>

<http://www.computerscienceonline.org/cs-programs-before-college/>

Content Area: Computer Science	
Unit Title: Computer Programming I - Unit 8 - Characters, Strings, and StringBuilder	
Grade Level: 10-12	
<p>Unit Summary: This unit covers working with characters strings in Java. Students will learn to use the Character, String, and StringBuilder classes. The Character class provides methods for working with single characters. The String class is most commonly used to represent a character string and is immutable. The StringBuilder class provides a mutable representation of a character string.</p> <p>Interdisciplinary Connections: Math: integers, decimals, Boolean numbers and other mathematical referenced and connections. English: connections to basic forms of speech (nouns, verbs, adjectives), and vocabulary.</p> <p>21st Century Themes and Skills: Creativity and Innovation, Communication and Collaboration, Critical Thinking & Problem Solving, Information, Media, and Technology Skills, Life and Career Skills.</p> <p>CRP1. Act as a responsible and contributing citizen and employee.</p> <p>CRP2. Apply appropriate academic and technical skills.</p> <p>CRP4. Communicate clearly and effectively</p> <p>CRP6. Demonstrate creativity and innovation.</p> <p>CRP8. Utilize critical thinking to make sense of problems and persevere in solving them.</p> <p>CRP11. Use technology to enhance productivity.</p>	
Standards (Content and Technology):	
CPI#:	Statement:
8.1.12.A.1	Create a personal digital portfolio which reflects personal and academic interests, achievements, and career aspirations by using a variety of digital tools and resources.
8.1.12.A.3	Collaborate in online courses, learning communities, social networks or virtual worlds to discuss a resolution to a problem or issue.
8.1.12.B.2	Apply previous content knowledge by creating and piloting a digital learning game or tutorial.
8.1.12.F.1	Evaluate the strengths and limitations of emerging technologies and their impact on educational, career, personal and or social needs.
8.2.12.E.1	Demonstrate an understanding of the problem-solving capacity of computers in our world.
8.2.12.E.3	Use a programming language to solve problems or accomplish a task (e.g., robotic functions, website designs, applications, and games
8.2.12.E.4	Use appropriate terms in conversation (e.g., troubleshooting, peripherals, diagnostic software, GUI, abstractions, variables, data types and conditional statements).
Unit Essential Question(s): <ul style="list-style-type: none"> How are characters strings used in Java? 	Unit Enduring Understandings: <ul style="list-style-type: none"> Students will learn to use the Character, String and StringBuilder classes
Unit Learning Targets/Objectives: <i>Students will</i> <ul style="list-style-type: none"> Identify string data problems Use Character class methods Declare and compare String objects Use other String methods Use the StringBuilder and StringBuffer classes 	

Formative Assessments:

- Teacher observations
- Practice programs
- Discussions

Summative/Benchmark Assessment(s):

- Checkpoint questions/quizzes
- Unit tests
- Programming Projects
- Chapter Portfolio
- Current Event Journal

Resources/Materials (copy hyperlinks for digital resources):

- Text Book: *Java Programming by Joyce Farrell*
- <https://login.cengage.com/cb/login.htm>
- www.codehs.com

Modifications:
Special Education Students

- Allow errors
- Rephrase questions, directions, and explanations
- Allow extended time to answer questions, and permit drawing, as an explanation
- Accept participation at any level, even one word
- Consult with Case Managers and follow IEP accommodations/modifications

At-Risk Students

- Provide extended time to complete tasks
- Consult with Guidance Counselors and follow I&RS procedures/action plans
- Consult with classroom teacher(s) for specific behavior interventions
- Provide rewards as necessary

English Language Learners

- Assign a buddy, same language or English speaking
- Allow errors in speaking
- Rephrase questions, directions, and explanations
- Allow extended time to answer questions
- Accept participation at any level, even one word

Gifted and Talented Students

- Provide extension activities
- Build on students' intrinsic motivations
- Consult with parents to accommodate students' interests in completing tasks at their level of engagement

Lesson Name/ Topic	Lesson Objective(s)	Time frame (day(s) to complete)
Under-standing String Data Problems	● To understand string data problems	½ day
Using Character Class Methods	● Learn to use the character class method	½ day
Declaring and Comparing	● Learn how to declare string objects	2 ½ days

String Objects	<ul style="list-style-type: none"> ● Learn how to compare string objects ● Learn about empty and null strings 	
Use other String Methods	<ul style="list-style-type: none"> ● Learn about the other string methods that are available. ● Learn how to convert String objects to numbers 	3 days
Learning about the StringBuilder and StringBuffer Classes	<ul style="list-style-type: none"> ● Learn about the StringBuilder class ● Learn about the StringBuffer class 	2 ½ days
Chapter Portfolio	<ul style="list-style-type: none"> ● Create a chapter portfolio containing vocabulary, reviews, and programming/debugging exercises. 	5 days
Programs	<ul style="list-style-type: none"> ● Game Program ● Case Study 	5 days

Teacher Notes: Program time may vary depending on the complexity of the programs given.

Additional Resources

Click links below to access additional resources used to design this unit:

<http://csunplugged.com/>

<https://code.org/>

<http://www.i-programmer.info/news.html>

<http://www.computerscienceonline.org/cs-programs-before-college/>

Content Area: Computer Science	
Unit Title: Computer Programming I - Unit 9 - Arrays	
Grade Level: 10-12	
<p>Unit Summary: This unit introduces the concept of arrays. An array is a list of elements of the same data type. Students will learn to create arrays of primitive data types and objects. They will work with arrays by searching, sorting, and passing them to methods.</p> <p>Interdisciplinary Connections: Math: integers, decimals, Boolean numbers and other mathematical referenced and connections. English: connections to basic forms of speech (nouns, verbs, adjectives), and vocabulary.</p> <p>21st Century Themes and Skills: Creativity and Innovation, Communication and Collaboration, Critical Thinking & Problem Solving, Information, Media, and Technology Skills, Life and Career Skills.</p> <p>CRP1. Act as a responsible and contributing citizen and employee.</p> <p>CRP2. Apply appropriate academic and technical skills.</p> <p>CRP4. Communicate clearly and effectively</p> <p>CRP6. Demonstrate creativity and innovation.</p> <p>CRP8. Utilize critical thinking to make sense of problems and persevere in solving them.</p> <p>CRP11. Use technology to enhance productivity.</p>	
Standards (Content and Technology):	
CPI#:	Statement:
8.1.12.A.1	Create a personal digital portfolio which reflects personal and academic interests, achievements, and career aspirations by using a variety of digital tools and resources.
8.1.12.A.3	Collaborate in online courses, learning communities, social networks or virtual worlds to discuss a resolution to a problem or issue.
8.1.12.B.2	Apply previous content knowledge by creating and piloting a digital learning game or tutorial.
8.1.12.F.1	Evaluate the strengths and limitations of emerging technologies and their impact on educational, career, personal and or social needs.
8.2.12.E.1	Demonstrate an understanding of the problem-solving capacity of computers in our world.
8.2.12.E.3	Use a programming language to solve problems or accomplish a task (e.g., robotic functions, website designs, applications, and games
8.2.12.E.4	Use appropriate terms in conversation (e.g., troubleshooting, peripherals, diagnostic software, GUI, abstractions, variables, data types and conditional statements).
Unit Essential Question(s): <ul style="list-style-type: none"> What is an array? How is an array used? 	Unit Enduring Understandings: <ul style="list-style-type: none"> Students will learn to create arrays of primitive data types and objects.
Unit Learning Targets/Objectives: <i>Students will</i> <ul style="list-style-type: none"> Declare arrays Initialize an array Use variable subscripts with an array Declare and use arrays of objects Search and array and use parallel arrays Pass arrays to and return arrays from methods 	

Formative Assessments:

- Teacher observations
- Practice programs
- Discussions

Summative/Benchmark Assessment(s):

- Checkpoint questions/quizzes
- Unit tests
- Programming Projects
- Chapter Portfolio
- Current Event Journal

Resources/Materials (copy hyperlinks for digital resources):

- Text Book: *Java Programming by Joyce Farrell*
- <https://login.cengage.com/cb/login.htm>
- www.codehs.com

Modifications:
Special Education Students

- Allow errors
- Rephrase questions, directions, and explanations
- Allow extended time to answer questions, and permit drawing, as an explanation
- Accept participation at any level, even one word
- Consult with Case Managers and follow IEP accommodations/modifications

At-Risk Students

- Provide extended time to complete tasks
- Consult with Guidance Counselors and follow I&RS procedures/action plans
- Consult with classroom teacher(s) for specific behavior interventions
- Provide rewards as necessary

English Language Learners

- Assign a buddy, same language or English speaking
- Allow errors in speaking
- Rephrase questions, directions, and explanations
- Allow extended time to answer questions
- Accept participation at any level, even one word

Gifted and Talented Students

- Provide extension activities
- Build on students' intrinsic motivations
- Consult with parents to accommodate students' interests in completing tasks at their level of engagement

Lesson Name/ Topic	Lesson Objective(s)	Time frame (day(s) to complete)
Declaring Arrays	● Learn how to declare an array	2 days
Initializing an Array	● Learn how to initialize an array	1 day
Using Variable subscripts with an Array	<ul style="list-style-type: none"> ● Learn how to use variable subscripts with an array ● Learn how to use the enhanced for loop ● Understand how to use part of an array 	2 days

Declaring and Using Arrays of Objects	<ul style="list-style-type: none"> ● Learn how to declare arrays of objects ● Learn how to use arrays of objects ● Learn how to use the enhanced for loop with objects ● Learn about manipulating arrays of Strings 	2 days
Searching an Array and Using Parallel Arrays	<ul style="list-style-type: none"> ● Learn how to search an array ● Learn how to use parallel arrays ● Understand how to search an array for a range match 	2 days
Passing Arrays to and Returning Arrays from Methods	<ul style="list-style-type: none"> ● Learn how to pass an array to and from methods. ● Learn how to return an array from a method. 	2 days
Chapter Portfolio	<ul style="list-style-type: none"> ● Create a chapter portfolio containing vocabulary, reviews, and programming/debugging exercises. 	5 days
Programs	<ul style="list-style-type: none"> ● Game Program ● Case Study 	5 days

Teacher Notes: Program time may vary depending on the complexity of the programs given.

Additional Resources

Click links below to access additional resources used to design this unit:

<http://csunplugged.com/>

<https://code.org/>

<http://www.i-programmer.info/news.html>

<http://www.computerscienceonline.org/cs-programs-before-college/>