West Windsor-Plainsboro Regional School District
Introduction to Computer Programming/Mobile App Design

Unit 1: Introduction to Computers and Programming

Content Area: Computer Science

Course & Grade Level: Introduction to Computer Programming/Mobile App Design, Grades 9 - 12

Summary and Rationale

This introductory unit covers the basics of computer systems, hardware and software, the concept of programming languages, elements of a program and the programming process.

Recommended Pacing

5 days

State Standards

Standard 8.1 Educational Technology

CPI #	Cumulative Progress Indicator (CPI)
A. Technology Operations and Concepts	
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.2	Produce and edit a multi-page digital document for a commercial or professional audience and present it to peers and/or professionals in that related area for review.
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.
Standard 8.2 Technology Education, Engineering, Design, and Computational Thinking - Programming	
CPI #	**Cumulative Progress Indicator (CPI)**
B. Technology and Society	
8.2.12.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.
C. Design	
8.2.12.C.1	Explain how open source technologies follow the design process.
8.2.12.C.2	Analyze a product and how it has changed or might change over time to meet human needs and wants.
8.2.12.C.4	Explain and identify interdependent systems and their functions.

E. Computational Thinking: Programming	
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.

9.3 – Career & Technical Education (CTE) Information Technology Career Cluster	
CPI #	**Cumulative Progress Indicator (CPI)**
Pathway: Programming & Software Development (IT-PRG)	
9.3.IT.12	Demonstrate knowledge of the hardware components associated with information systems.
9.3.IT-PRG.3	Analyze system and software requirements to ensure maximum operating efficiency.

Instructional Focus

Unit Enduring Understandings

- The goal of the Introduction to Computers and Programming Unit is to develop an understanding of the basic ideas of hardware/software and the concepts we will use throughout the course.
- There are many programming languages.
- There are certain elements that are common to all programming languages.
- Elements of a program: keywords, operators, punctuation, identifiers, syntax, variables
- Use of a compiler and virtual machine.
- The programming process consists of design, creation, testing and debugging.
- Object oriented programming is a method method of software development that has its own practices, concepts and vocabulary.

Unit Essential Questions

- What is hardware?
- What is software?
- What are some programming languages?
- What are the elements of a program?
- How is a compiler and virtual machine used?
- What are the steps in the programming process?
- Why are there steps in the programming process?
- What is object-oriented programming?

Objectives
Students will know: ● Examples of hardware and software ● How hardware and software interact ● The names of several programming languages ● The elements of a program ● How to use a compiler and virtual machine ● How to describe object oriented programming ● How to write, compile and run simple programs **Students will be able to:** ● Provide examples of hardware and software ● Describe how hardware and software interact ● Name several programming languages ● Define the elements of a program ● Describe the steps in the programming process and the purpose of each ● Use a compiler and virtual machine ● Describe object oriented programming
Resources
Core Text: Starting out with Java: Early Objects. Tony Gaddis. Addison-Wesley. Chapter 1. Suggested Resources: http://cpsr.org/issues/ethics/cei/

Unit 2: Introduction to Object Oriented Programming

Content Area: Computer Science

Course & Grade Level: Introduction to Computer Programming/Mobile App Design, Grades 9 - 12

Summary and Rationale

Every Java program has specific parts and utilizes fundamental data types. In this unit students explore the parts of a program and learn to manipulate types of data.

Recommended Pacing

10 days

State Standards

Standard 8.1 Educational Technology

CPI #	Cumulative Progress Indicator (CPI)
B. Creativity and Innovation	
8.1.12.B.2	Apply previous content knowledge by creating and piloting a digital learning game or tutorial.

Standard 8.2 Technology Education, Engineering, Design, and Computational Thinking - Programming

CPI #	Cumulative Progress Indicator (CPI)
E. Computational Thinking: Programming	
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, abstraction, variables, data types and conditional statements).

9.3 – Career & Technical Education (CTE) Information Technology Career Cluster

CPI #	Cumulative Progress Indicator (CPI)
Pathway: Programming & Software Development (IT-PRG)	
9.3.IT.7	Perform standard computer backup and restore procedures to protect IT information.
9.3.IT-PRG.6	Program a computer application using the appropriate programming language.
9.3.IT-PRG.7	Demonstrate software testing procedures to ensure quality products.

Instructional Focus
Unit Enduring Understandings
● The goal of the unit is to explore the basic elements of a Java program.
Unit Essential Questions
● What are the basic parts of a Java program? ● What purpose does each part of the program serve? ● What is a variable? ● What is a literal? ● What is an identifier? ● What are the primitive data types? ● What makes them primitive? ● How can we evaluate and manipulate these data types? ● What is the scope of a variable? ● How and why is a program properly documented? ● In what ways can input be read into the computer from the keyboard? ● How can output be formatted from a program so that it is aesthetically pleasing to the user?
Objectives
Students will know: ● How to represent data using a variable. ● How to evaluate/simplify numerical expressions. ● How to manipulate String objects. ● How to enter data from the keyboard generically and using dialog boxes. ● How to properly document a program. ● How to research using the Java API. ● How to format output in a professional manner. **Students will be able to:** ● Identify the parts of a program. ● Utilize the Java API. ● Declare, define and utilize variables appropriately. ● Evaluate Java numerical expressions. ● Manipulate Strings. ● Convert between primitive data types. ● Input data from the keyboard. ● Input data from dialog boxes. ● Format output.

Resources
Core Text: Starting out with Java: Early Objects. Tony Gaddis. Addison-Wesley. Chapter 2. **Suggested Resources:** http://docs.oracle.com/javase/6/docs/api/

Unit 3: A First Look at Classes and Objects

Content Area: Computer Science

Course & Grade Level: Introduction to Computer Programming/Mobile App Design, Grades 9 - 12

Summary and Rationale

This unit provides an introduction to Classes and Objects.

Recommended Pacing

20 days

State Standards

Standard 8.1 Educational Technology

CPI #	Cumulative Progress Indicator (CPI)
B. Creativity and Innovation	
8.1.12.B.2	Apply previous content knowledge by creating and piloting a digital learning game or tutorial.

Standard 8.2 Technology Education, Engineering, Design, and Computational Thinking - Programming

CPI #	Cumulative Progress Indicator (CPI)
E. Computational Thinking: Programming	
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, abstraction, variables, data types and conditional statements).

9.3 – Career & Technical Education (CTE) Information Technology Career Cluster

CPI #	Cumulative Progress Indicator (CPI)
Pathway: Programming & Software Development (IT-PRG)	
9.3.IT.2	Use product or service design processes and guidelines to produce a quality information technology (IT) product or service.
9.3.IT.7	Perform standard computer backup and restore procedures to protect IT information.
9.3.IT-PRG.6	Program a computer application using the appropriate programming language.
9.3.IT-PRG.7	Demonstrate software testing procedures to ensure quality products.

Instructional Focus

Unit Enduring Understandings

- The goals of Unit 3 are to understand:
  - Classes and Objects, or instances, of classes.
  - Attributes and Methods
  - The use of constructors
  - Importing packages

Unit Essential Questions

- How do I plan and build a class?
- How do I create different objects/instances from a class?
- How do I use constructors, attributes and methods?
- How do I import packages?

Objectives

Students will know:

- How to define and create the code for a class.
- How to create objects or instances of a class, each with their own set of data.
- How to call a constructor, determine attributes and create methods to perform operations.
- How to import pre-written Java packages.

Students will be able to:

- Create classes.
- Create objects or different instances of a class.
- Use constructors, attributes and methods.
- Import packages.

Resources

Core Text: Starting out with Java: Early Objects. Tony Gaddis. Addison-Wesley. Chapter 3.

Suggested Resources: http://docs.oracle.com/javase/6/docs/api/

Unit 4: Decision Structures	
Content Area: Computer Science	
Course & Grade Level: Introduction to Computer Programming/Mobile App Design, Grade 9 - 12	
Summary and Rationale	
Any practical computer program contains many statements that make decisions. In this unit we examine selection statements and how they affect the flow of a program.	
Recommended Pacing	
20 days	
State Standards	
Standard 8.1 Educational Technology	
CPI #	**Cumulative Progress Indicator (CPI)**
B. Creativity and Innovation	
8.1.12.B.2	Apply previous content knowledge by creating and piloting a digital learning game or tutorial.
Standard 8.2 Technology Education, Engineering, Design, and Computational Thinking - Programming	
CPI #	**Cumulative Progress Indicator (CPI)**
E. Computational Thinking: Programming	
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, abstraction, variables, data types and conditional statements).
9.3 – Career & Technical Education (CTE) Information Technology Career Cluster	
CPI #	**Cumulative Progress Indicator (CPI)**
Pathway: Programming & Software Development (IT-PRG)	
9.3.IT.2	Use product or service design processes and guidelines to produce a quality information technology (IT) product or service.
9.3.IT.7	Perform standard computer backup and restore procedures to protect IT information.
9.3.IT-PRG.1	Analyze customer software needs and requirements.

9.3.IT-PRG.5	Apply an appropriate software development process to design a software application.
9.3.IT-PRG.6	Program a computer application using the appropriate programming language.
9.3.IT-PRG.7	Demonstrate software testing procedures to ensure quality products.
9.3.IT-PRG.8	Perform quality assurance tasks as part of the software development cycle.

Instructional Focus

Unit Enduring Understandings

- The goal of the unit is to learn how to program simple and complex decisions.

Unit Essential Questions

- How can we utilize decision making in a Java program?
- What is an if block?
- What are Boolean expressions?
- What is a switch?
- How can we compare primitive data types?
- How can we compare objects?

Objectives

Students will know:

- How to evaluate Boolean expressions
- How to write if statements and if-else blocks.
- How to write switch statements

Students will be able to:

- Implement selection control using if statements
- Implement selection control using if-else blocks.
- Write Boolean expressions
- Evaluate Boolean expressions.
- Implement selection control by using switch statements
- Generate random numbers and utilize them in applications.

Resources

Core Text: Starting out with Java: Early Objects. Tony Gaddis. Addison-Wesley. Chapter 4.

Suggested Resources: http://docs.oracle.com/javase/6/docs/api/

Unit 5: Iteration	
Content Area: Computer Science	
Course & Grade Level: Introduction to Computer Programming/Mobile App Design, Grades 9 - 12	
Summary and Rationale	
This unit features looping and file input/output.	
Recommended Pacing	
30 days	
State Standards	
Standard 8.1 Educational Technology	
CPI #	**Cumulative Progress Indicator (CPI)**
A. Technology Operations and Concepts	
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.
B. Creativity and Innovation	
8.1.12.B.2	Apply previous content knowledge by creating and piloting a digital learning game or tutorial.
Standard 8.2 Technology Education, Engineering, Design, and Computational Thinking - Programming	
CPI #	**Cumulative Progress Indicator (CPI)**
E. Computational Thinking: Programming	
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, abstraction, variables, data types and conditional statements).
9.3 – Career & Technical Education (CTE) Information Technology Career Cluster	
CPI #	**Cumulative Progress Indicator (CPI)**
Pathway: Programming & Software Development (IT-PRG)	
9.3.IT.2	Use product or service design processes and guidelines to produce a quality information technology (IT) product or service.
9.3.IT.7	Perform standard computer backup and restore procedures to protect IT information.

9.3.IT-PRG.1	Analyze customer software needs and requirements.
9.3.IT-PRG.2	Demonstrate the use of industry standard strategies and project planning to meet customer specifications.
9.3.IT-PRG.3	Analyze system and software requirements to ensure maximum operating efficiency.
9.3.IT-PRG.4	Demonstrate the effective use of software development tools to develop software applications.
9.3.IT-PRG.5	Apply an appropriate software development process to design a software application.
9.3.IT-PRG.6	Program a computer application using the appropriate programming language.
9.3.IT-PRG.7	Demonstrate software testing procedures to ensure quality products.
9.3.IT-PRG.8	Perform quality assurance tasks as part of the software development cycle.

Instructional Focus

Unit Enduring Understandings

- Understand and utilize the increment and decrement operators
- Understand the following loop structures: while loop, do-while loop, for loop as well as nested loops
- Use the while loop for input validation
- Implement Running Totals and Sentinel Values
- Effectively utilize break and continue statements
- Introduction to File Input and Output

Unit Essential Questions

- Why are loops used in programming?
- What are the different types of loop structures?
- What are Running Totals and Sentinel Values?
- How and Why are Break and Continue Statements used?
- How can the programmer decide which loops, increments, and operators to use?
- Why is file Input and Output important?

Objectives
Students will know: ● How to use Increment and Decrement Operators ● How to use the following: while loop, do-while loop, for loop, nested loops ● How to implement input validation techniques ● How to create Running Totals and Sentinel Values ● How to use Break and Continue Statements ● How to use File Input to process/display information and File Output to process/write to data files. **Students will be able to:** ● Create programs with Increment and Decrement Operators ● Create programs with various looping structures ● Implement validation techniques ● Create Running Totals and Sentinel Values ● Use Break and Continue Statements ● Write programs incorporating File Input to process/display information and File Output to process/write to data files.
Resources
Core Text: Starting out with Java: Early Objects. Tony Gaddis. Addison-Wesley. Chapter 5. **Suggested Resources:** http://docs.oracle.com/javase/6/docs/api/

Unit 6: Arrays

Content Area: Computer Science

Course & Grade Level: Introduction to Computer Programming/Mobile App Design, Grades 9 - 12

Summary and Rationale

Arrays can be a powerful data structure in computer science. We will explore how to create and use them.

Recommended Pacing

20 days

State Standards

Standard 8.1 Educational Technology

CPI #	Cumulative Progress Indicator (CPI)
A. Technology Operations and Concepts	
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.
B. Creativity and Innovation	
8.1.12.B.2	Apply previous content knowledge by creating and piloting a digital learning game or tutorial.

Standard 8.2 Technology Education, Engineering, Design, and Computational Thinking - Programming

CPI #	Cumulative Progress Indicator (CPI)
E. Computational Thinking: Programming	
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, abstraction, variables, data types and conditional statements).

9.3 – Career & Technical Education (CTE) Information Technology Career Cluster

CPI #	Cumulative Progress Indicator (CPI)
Pathway: Programming & Software Development (IT-PRG)	
9.3.IT.2	Use product or service design processes and guidelines to produce a quality information technology (IT) product or service.

9.3.IT.7	Perform standard computer backup and restore procedures to protect IT information.
9.3.IT-PRG.1	Analyze customer software needs and requirements.
9.3.IT-PRG.2	Demonstrate the use of industry standard strategies and project planning to meet customer specifications.
9.3.IT-PRG.3	Analyze system and software requirements to ensure maximum operating efficiency.
9.3.IT-PRG.4	Demonstrate the effective use of software development tools to develop software applications.
9.3.IT-PRG.5	Apply an appropriate software development process to design a software application.
9.3.IT-PRG.6	Program a computer application using the appropriate programming language.
9.3.IT-PRG.7	Demonstrate software testing procedures to ensure quality products.
9.3.IT-PRG.8	Perform quality assurance tasks as part of the software development cycle.

Instructional Focus

Unit Enduring Understandings

- The goal of the unit is to explore and apply the basic algorithms of arrays.

Unit Essential Questions

- What is an Array?
- How do we put data into an Array?
- How do we access data stored in an array?
- How can we utilize arrays to make programs more efficient?

Objectives
Students will know: ● How to create an array of numbers/Strings. ● How to traverse through an array. ● How to pass arrays as arguments. ● How to use basic array algorithms. **Students will be able to:** ● Create an array of numbers. ● Traverse through an array of numbers. ● Search and sort an array of numbers or strings ● Find the min/max/sum/average of an array of numbers. ● Create an array of Strings. ● Traverse through an array of Strings. ● Access data stored in an array.
Resources
Core Text: Starting out with Java: Early Objects. Tony Gaddis. Addison-Wesley. Chapter 7. **Suggested Resources:** http://docs.oracle.com/javase/6/docs/api/

Unit 7: Mobile App Design

Content Area: Computer Science

Course & Grade Level: Introduction to Computer Programming/Mobile App Design, Grades 9 - 12

Summary and Rationale

In providing students with the skills that developers use when conceptualizing and creating mobile apps, and with the goals of expanding students' problem-solving abilities, critical-thinking processes and technical programming skills, this unit serves as the capstone for the Introduction to Programming course.

Recommended Pacing

30 days

State Standards

Standard 8.1 Educational Technology

CPI #	Cumulative Progress Indicator (CPI)
A. Technology Operations and Concepts	
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.2	Produce and edit a multi-page digital document for a commercial or professional audience and present it to peers and/or professionals in that related area for review.
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.
B. Creativity and Innovation	
8.1.12.B.2	Apply previous content knowledge by creating and piloting a digital learning game or tutorial.
C. Communication and Collaboration	
8.1.12.C.1	Develop an innovative solution to a real world problem or issue in collaboration with peers and experts, and present ideas for feedback through social media or in an online community.
D. Digital Citizenship	
8.1.12.D.1	Demonstrate appropriate application of copyright, fair use and/or Creative Commons to an original work.

Standard 8.2 Technology Education, Engineering, Design, and Computational Thinking - Programming	
CPI #	**Cumulative Progress Indicator (CPI)**
D. Abilities for a Technological World	
8.2.12.D.1	Design and create a prototype to solve a real world problem using a design process, identify constraints addressed during the creation of the prototype, identify trade-offs made, and present the solution for peer review.
E. Computational Thinking: Programming	
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, abstraction, variables, data types and conditional statements).
9.3 – Career & Technical Education (CTE) Information Technology Career Cluster	
CPI #	**Cumulative Progress Indicator (CPI)**
Pathway: Programming & Software Development (IT-PRG)	
9.3.IT.1	Demonstrate effective professional communication skills and practices that enable positive customer relationships.
9.3.IT.2	Use product or service design processes and guidelines to produce a quality information technology (IT) product or service.
9.3.IT.7	Perform standard computer backup and restore procedures to protect IT information.
9.3.IT-PRG.1	Analyze customer software needs and requirements.
9.3.IT-PRG.2	Demonstrate the use of industry standard strategies and project planning to meet customer specifications.
9.3.IT-PRG.3	Analyze system and software requirements to ensure maximum operating efficiency.
9.3.IT-PRG.4	Demonstrate the effective use of software development tools to develop software applications.
9.3.IT-PRG.5	Apply an appropriate software development process to design a software application.
9.3.IT-PRG.6	Program a computer application using the appropriate programming language.
9.3.IT-PRG.7	Demonstrate software testing procedures to ensure quality products.
9.3.IT-PRG.8	Perform quality assurance tasks as part of the software development cycle.

Instructional Focus

Unit Enduring Understandings

The goal of the unit is to implement the design process: Ideate, Plan, Calculate and Build. This process guides students in a logical order and identifies the steps of a complex procedure with the intent of developing good habits for problem solving, utilizing mobile app development as the platform.

Unit Essential Questions

- What are the steps in the design process?
- How do we define our ideas?
- How do we build our ideas?
- How do we test our ideas?
- How do we present our ideas?

Objectives

Students will know:

- Industry mobile app-developer documentation format and terms.
- To consider audience and user profiles for original app concepts.
- How to code a mobile app based on client specifications.
- How to test, evaluate, modify and reflect on the app they have created.

Students will be able to:

- Read and comprehend technically challenging information from the field of mobile app development.
- Gain, evaluate and present complex information, ideas and evidence through speaking and listening.
- Present the concept for development.
- Follow complex instructions to learn to use a SDK to build mobile applications.
- Create solutions in the form of mobile apps.

Resources

Core Text: Android App Inventor for the Absolute Beginner. Prayaga, Hawthorne and Whiteside. Cengage Learning.

Suggested Resources: http://appinventor.mit.edu/explore/