

HS Computer Science I: App Development

Huntingdon Area School District

UNITS (4/4 SELECTED)

- Unit 1: Values
- Unit 2: Algorithms
- Unit 3: Organizing Data
- Unit 4: Building Apps

SUGGESTED DURATION

- 4 lessons*
- 5 lessons*
- 6 lessons*
- 4 lessons*

Unit 1: Values

HS Computer Science I: App Development

UNIT OVERVIEW

Students will learn about and use values that flow through their code, including text and numbers. Students will explore how to associate names with values using variables and constants, and how values can be used in simulations.

STANDARDS/EXPECTATIONS

Pennsylvania - Grade 9-10 - Computer Science

3A.CS.02

3A.DA.11

3A.AP.17

3A.AP.18

3A.AP.23

Pennsylvania - Grade 11-12 - Computer Science

3B.AP.13

3B.AP.16

3B.AP.19

BIG IDEAS

Big Ideas

- Information in a computer/device is stored as data
- Values can be stored as a variable or a constant
- Devices contain systems that have data inputs and outputs
- Written code will be run through Apples Interface Builder

ESSENTIAL QUESTIONS

Essential Questions

- How is data collected, stored, and used in computer programs?
- What is the difference between values that are stored as a constant versus a variable?
- What kind of identifiers should be used when naming data?

Unit 1: Values

HS Computer Science I: App Development

LEARNING TARGETS: KNOWLEDGE & SKILLS

Knowledge	Skills
Students will know (Acquired Knowledge)	Students can do (Acquired Skill)
how changing the code that is input into the program will affect the output of the program.	the playground basics file and will experiment with running code in Xcode 12.
how to locate errors in code and will be able to debug and fix the issue(s).	will write code, watch it run, alter code, and see how the output changes.
the fundamentals of solving problems by using the good names and identifiers as a basis for their work.	declare a new constant using the "let" keyword, choose a meaningful, specific name for a value, and refer to a previously declared value by its name.
learn how simulations use values to define the behavior of the model.	describe how a simulation's behavior depends on its parameters, will interpret a simulation's visual information, and hypothesize how a simulation might change when one of its parameters is adjusted.
expand and apply an understanding of naming as they're introduced to the concepts of constants and variables.	describe the difference between constants and variables and identify appropriate times to use variables over constants.
what a string is and how they are used to store text, or a series of characters.	identify different types of characters, declare new strings, combine existing strings, and use the escape characters to output special text.

Unit 1: Values

HS Computer Science I: App Development

EVIDENCE OF LEARNING & ASSESSMENT

Name of Assessment	Type (formative, summative, project-based, diagnostic)	Description
Naming and Identifying Activity	Formative	Students will use a photo of an everyday object and use Markup to identify as many different physical attributes of that object as they can.
Xcode "Playgrounds"	Summative	Students will work through and complete all of the following Playgrounds: Basics, Naming and Identifiers, Simulation, Constants and Variables, and Strings.
Word Games	Project Based	Students will "build" a fill-in-the-blank word game using code to express themselves creatively.
Photo Frame App	Project Based	Students will use Xcode to create their first app which will be a Photo Frame.

Unit 2: Algorithms

HS Computer Science I: App Development

UNIT OVERVIEW

Students will learn about the fundamentals of structured code. They will explore how functions can encapsulate repetitive tasks and how to use if/else statement to make decisions in their code. They will also learn how Swift language uses types to distinguish different kinds of data.

STANDARDS/EXPECTATIONS

Pennsylvania - Grade 9-10 - Computer Science

3A.CS.01

3A.CS.02

3A.AP.13

3A.AP.14

3A.AP.15

3A.AP.18

3A.AP.23

Pennsylvania - Grade 11-12 - Computer Science

3B.AP.10

3B.AP.11

3B.AP.22

3B.AP.23

BIG IDEAS

Big Ideas

- Procedural abstraction is pivotal to coding and will be used every time they develop an algorithm to solve a problem
- Developing an algorithm then finding the right procedure is necessary to accomplish their goal
- Sequencing, selection, and iteration are the three algorithmic building blocks
- Simplifying code can be done using functions, types, parameters and Boolean logic.

ESSENTIAL QUESTIONS

Essential Questions

- How can you define inputs that can be used repetitively?
- How does order and the decision making within code affect the corresponding output?

Unit 2: Algorithms

HS Computer Science I: App Development

LEARNING TARGETS: KNOWLEDGE & SKILLS

Knowledge	Skills
Students will know (Acquired Knowledge)	Students can do (Acquired Skill)
what makes functions so powerful, as they combine detailed steps into a definition that can be used again and again.	describe why functions are an important concept and implement basic functions.
more about the type system and Swift library.	understand and explain a variety of types from the Swift standard library and identify cases where the Swift type system will infer a symbol's type.
how to expand functions by using parameters and return values to make functions more flexible and powerful.	create a function that accepts and uses parameters, create a function that returns a value, and uses appropriate naming conventions to clearly describe functions and parameters.
how to make decisions in code with conditional if/else statements, true or false values, and comparison operators.	compare values as well as understand, explain and use conditionals to write code that will run specific lines of code based on certain values.

Unit 2: Algorithms

HS Computer Science I: App Development

EVIDENCE OF LEARNING & ASSESSMENT

Name of Assessment	Type (formative, summative, project-based, diagnostic)	Description
Robot Artist	Formative	Students will take turns giving clear, concise commands. Students will pair up and take turns being the "robot programmer" and "robot artist" with another student in the class.
Sequencing Photos	Formative	For most tasks, the order of instructions is important (often crucial). Take a series of photos and arrange them to illustrate a sequence of actions.
Xcode Playgrounds	Summative	<div style="border: 1px solid black; padding: 5px;">Students will work through, complete, and share-out all of the following Playgrounds: Functions, Types, Parameters and Results, and Making Decisions.</div>
Boogie Bot	Project-Based	Students will write dance code and functions for a robot.
Question Bot	Project-Based	Students will modify existing code and write new logic for a chatbot.

Unit 3: Organizing Data

HS Computer Science I: App Development

UNIT OVERVIEW

Students will learn skills for organizing and processing data. Students will explore how to create custom types with structs, group large quantities of items into arrays, and process them using loops. Students will conclude this unit by building an interactive game app with colorful shapes.

STANDARDS/EXPECTATIONS

Pennsylvania - Grade 11-12 - Computer Science

3B.DA.07

3B.AP.10

3B.AP.11

3B.AP.12

3B.AP.13

3B.AP.23

Pennsylvania - Grade 9-10 - Computer Science

3A.NI.07

3A.DA.10

3A.AP.15

3A.AP.17

3A.AP.18

3A.AP.21

3A.IC.29

BIG IDEAS

Big Ideas

- Types enable you to understand what each value in your code represents
- Lists are a way of organizing information by arranging items in order.
- **There are different ways to repeat steps in an algorithm known as iteration which most commonly is referred to as a loop**
- **There are different ways of finding an item in an array, known as searching**

ESSENTIAL QUESTIONS

Essential Questions

- How can you form and arrange a list of information or items?
- How can you repeat specific steps in an algorithm?
- What are different ways to search for instances or items in your code?

Unit 3: Organizing Data

HS Computer Science I: App Development

LEARNING TARGETS: KNOWLEDGE & SKILLS

Knowledge	Skills
Students will know (Acquired Knowledge)	Students can do (Acquired Skill)
that types allow you to understand what each value in your code represents, and type safety makes sure you use values of the right type in the right place.	<ul style="list-style-type: none">• create values without using literals• describe the difference between a type and an instance• describe the difference between a method and a property
arrays provide a common way to group objects in an ordered list. Students will learn how to create and work with arrays by adding and removing objects. Students will also learn how to use a for... in loop to work with each object in an array.	<ul style="list-style-type: none">• develop data abstraction using array to store multiple elements• explain how the use of data abstraction manages complexity in program code• use arrays to manage collections of objects• iterate through an array to perform a common action on each item in the collection
recognize that it is often useful to group related information and functionality into a custom type. Students will learn how to define a custom type by using a structure with accompanying properties and methods.	<ul style="list-style-type: none">• explain the importance of custom structures and their common use cases.• design a custom structure to group related data into one type• define custom properties and methods for custom structures
enumerations (enums) are a way to define a named list of options. Students will learn what enums are used for, how to define them, and common ways to work with them. Students will also learn to use the switch statement to conditionally run blocks of code based on the value of an enum instance.	<ul style="list-style-type: none">• explain the benefits and proper use cases of enumerations.• use a switch statement on an enum value to conditionally trigger specific code.
how to evaluate the way functions might fail, and think about the inputs and outputs of functions.	<ul style="list-style-type: none">• evaluate functions and determine valid and invalid outputs and inputs

Unit 3: Organizing Data

HS Computer Science I: App Development

Knowledge	Skills
	<ul style="list-style-type: none">• write tests for functions that cover a range of possible values, including ones that will fail• describe the limitations of the built-in integer and floating-point types
<p>how to process a large data set from a hypothetical survey. They'll gain an appreciation for how errors can occur in data sets and learn strategies for correcting them. Students will also encounter third-party code and use it to solve a problem which is universal practice.</p>	<ul style="list-style-type: none">• tally basic information in a large data set• identify and correct simple error using a variety of data transformation techniques• translate data to derive statistical information

Unit 3: Organizing Data

HS Computer Science I: App Development

EVIDENCE OF LEARNING & ASSESSMENT

Name of Assessment	Type (formative, summative, project-based, diagnostic)	Description
Xcode Playgrounds	Summative	Students will work through, complete, and share-out all of the following Playgrounds: Instance, Methods & Playgrounds, Arrays & Loops, Structures, Enums & Switch, Testing Code, Processing Data, Pixel Art, Password Security, and Visualization Revisited.
Bouncy Ball App	Project Based	Students will modify existing code and write new code for a game involving a bouncing ball and a target.

Unit 4: Building Apps

HS Computer Science I: App Development

UNIT OVERVIEW

Students deepen their skills in Xcode and Interface Builder in guided projects that show how to build an app from the ground up. Students use techniques to add user interface elements to a screen, connect those elements to their code, and respond to the events generated by user interaction. They also experience the incremental development of an app by gradually expanding its capabilities, testing it at each phase.

STANDARDS/EXPECTATIONS

Pennsylvania - Grade 9-10 - Computer Science

3A.CS.02

3A.CS.03

3A.DA.10

3A.AP.17

3A.AP.19

3A.AP.21

3A.IC.26

Pennsylvania - Grade 11-12 - Computer Science

3B.AP.11

3B.AP.13

3B.AP.16

3B.AP.21

BIG IDEAS

Big Ideas

- Understand how the app should behave.
- Anticipate all the possible paths a user could take through the app—not just the most common ones.
- Test all these paths, or at least a representative sample of them.
- For the paths that fail, report the failure in detail: the steps it takes to reproduce the error, how the error manifests, and how reliably it can be reproduced.

ESSENTIAL QUESTIONS

Essential Questions

- What kinds of inputs do commonly used apps contain?
- What kinds of outputs do commonly used apps produce?
- What are all the different situations that the app designer might have had to account for when designing it?

Unit 4: Building Apps

HS Computer Science I: App Development

LEARNING TARGETS: KNOWLEDGE & SKILLS

Knowledge	Skills
Students will know (Acquired Knowledge)	Students can do (Acquired Skill)
how to debug different error types	-identify and correct logic errors -identify and correct run-time errors -identify and correct
delegation is a common pattern for division of labor with an app.	-use delegation to have an app divide responsibility on how tasks are completed
apps often receive much of their input from events that they don't control.	Account for a user action such as a tap or swipe, or other actions such as an incoming phone call or data arriving from the network.

Unit 4: Building Apps

HS Computer Science I: App Development

EVIDENCE OF LEARNING & ASSESSMENT

Name of Assessment	Type (formative, summative, project-based, diagnostic)	Description
Color Picker	Formative	In this lesson, students will learn how to connect Swift code to the views and controls you create in a storyboard, so your code is able to respond to the user's actions.
Creating Switches	Project-based	In this part, students will add a UISwitch control. You'll define an action method and connect it to the switch.
Slider Controls	Project-based	In this part of the lesson, students will add slider controls—like the brightness or volume sliders on an iOS device—to provide finer control.
Polishing the Interface	Project-based	In this portion, students will generate reset buttons, allow color changes, and adding borders via outlets.
Meme Maker	Summative & Project-based	Students will review a variety of concepts covered in the course, and build the user interface, the model data, and the controller objects that make up the entire application. They'll also use a gesture recognizer to enable users to move elements on the

Unit 4: Building Apps

HS Computer Science I: App Development

Name of Assessment	Type (formative, summative, project-based, diagnostic)	Description
		screen.