

03 Navigation and Workflows

Content Area: **Math**
Course(s):
Time Period: **Semester**
Length: **6 - 8 Weeks**
Status: **Published**

General Overview, Course Description or Course Philosophy

Introduction to App Development is a semester course that introduces students to fundamental concepts of programming and application design through exercises, labs, and app projects.

Students will become familiar with the process of debugging and troubleshooting their code; develop new ways of thinking that can be applied outside the classroom. They will learn how to articulate their ideas, collaborate on large complicated projects and how to design with a purpose. Students will have the opportunity to foster their ability to develop and apply computational and design thinking to address real-world problems while design creative solutions. Students will engage not only independently but will work collaboratively to navigate the dynamic digital landscape.

In this unit, students will explore how to work with optional data, represent a set of related values with enums. Students will focus on writing safer code and reduce complicated logic by using optionals and guards. Students will explore iOS navigation paradigms by adding segues between scenes, pushing and posing content, using navigation controls such as tab and navigation bars to make the users experience less cumbersome. Lastly students will explore the life cycle of an application. The unit culminates with an application project.

OBJECTIVES, ESSENTIAL QUESTIONS, ENDURING UNDERSTANDINGS

Objectives:

- Understand the life cycle of an application
- Optimize user experience by using segues and navigation
- Use optionals and guards to create simpler and safer code that checks for nil-value checking

Essential Questions

- How does Swift take advantage of flexible variable types?
- How does nested logic alter the execution of code?
- How does designing a system of interacting modules manage the complexity of a program?
- How can users transition between scenes easily?
- How can you check for nil-values?

Enduring Understanding

- Navigation controllers, tabs, and workflow are used to control how content is organized.
- Swift allows for flexibility in variable creation and assignment.
- Swift has multiple syntax that carries out the same logic.
- Nil-value checking will allow for safer written code.

CONTENT AREA STANDARDS

CS.9-12.8.1.12.AP.1	Design algorithms to solve computational problems using a combination of original and existing algorithms.
CS.9-12.8.1.12.AP.2	Create generalized computational solutions using collections instead of repeatedly using simple variables.
CS.9-12.8.1.12.AP.3	Select and combine control structures for a specific application based upon performance and readability, and identify trade-offs to justify the choice.
CS.9-12.8.1.12.AP.4	Design and iteratively develop computational artifacts for practical intent, personal expression, or to address a societal issue.
CS.9-12.8.1.12.AP.5	Decompose problems into smaller components through systematic analysis, using constructs such as procedures, modules, and/or objects.

RELATED STANDARDS (Technology, 21st Century Life & Careers, ELA Companion Standards are Required)

LA.K-12.NJSLSA.L1	Demonstrate command of the conventions of standard English grammar and usage when writing or speaking.
LA.K-12.NJSLSA.L6	Acquire and use accurately a range of general academic and domain-specific words and phrases sufficient for reading, writing, speaking, and listening at the college and career readiness level; demonstrate independence in gathering vocabulary knowledge when encountering an unknown term important to comprehension or expression.
LA.K-12.NJSLSA.SL1	Prepare for and participate effectively in a range of conversations and collaborations with diverse partners, building on others' ideas and expressing their own clearly and persuasively.
LA.K-12.NJSLSA.SL2	Integrate and evaluate information presented in diverse media and formats, including visually, quantitatively, and orally.
MA.K-12.1	Make sense of problems and persevere in solving them.
MA.K-12.2	Reason abstractly and quantitatively.
MA.K-12.3	Construct viable arguments and critique the reasoning of others.
MA.K-12.5	Use appropriate tools strategically.
MA.K-12.6	Attend to precision.
MA.K-12.7	Look for and make use of structure.
MA.K-12.8	Look for and express regularity in repeated reasoning.
WRK.K-12.P.4	Demonstrate creativity and innovation.

WRK.K-12.P.5	Utilize critical thinking to make sense of problems and persevere in solving them.
WRK.K-12.P.8	Use technology to enhance productivity increase collaboration and communicate effectively.
WRK.K-12.P.9	Work productively in teams while using cultural/global competence.

STUDENT LEARNING TARGETS

Declarative Knowledge

Students will understand that:

- nil refers to a variable that contains nothing
- Optional is a value of either an underlying value or nil
- Checking for nil first will allow for simpler logic and safer code
- Structures are used to make working with complex data and performing logic easier
- Any and AnyObject is used to re any instance or a class of any type
- Although guard statements can be written an if statements, guard statements improve readability
- Variables only exist within their scope
- Enumerations are used to group related values with a common type
- Enumerations make code more readable and easier to maintain
- Navigation controllers are used to pushing and pop content in an app
- Tab bars organize content into a tab-based view hierarchy

Procedural Knowledge

Students will be able to

- Create an optional value
- Unwrap a variable
- Create a failable initializer
- Iterate through heterogeneous collections
- Demonstrate how to build a function that guards against invalid arguments
- Determine the scope of a variable
- Demonstrate how to define an enumeration with explicitly defined values
- Demonstrate how to use an enumeration in a switch statement
- Demonstrate the proper use of sequels to navigate between scenes
- Create a tab bar controller in a storyboard
- Identify the correct event in a view controller life cycle for executing setup logic

EVIDENCE OF LEARNING

Formative Assessments

- Guided practice
- Independent practice
- Check Lists
- Exit Ticket
- Class Discussion
- Teacher Observation
- Exit/Entrance Tickets
- Classwork
- Labs

Summative Assessments

- End of Unit Test
- Performance Task Examples:
 - Personality Quiz
 - Elements Quiz

RESOURCES (Instructional, Supplemental, Intervention Materials)

- Developing in Swift Fundamentals - Teachers Guide
- Developing in Swift Fundamentals - Student Book
- [Teacher Resources](#) (slides and answer key to labs and classwork)
- [Student Labs and Classwork](#)
- [Quick Reference Guide](#)

INTERDISCIPLINARY CONNECTIONS

Interdisciplinary connections are frequently addressed through modeling existing applications

and labs whereby creating solutions and analyzing situations from business, geography, health/fitness, statistics and numerous other fields. Examples can be found in topics specific textbook examples, lab practice, guided projects and digital resources.

ACCOMMODATIONS & MODIFICATIONS FOR SUBGROUPS

See link to Accommodations & Modifications document in course folder.