

App Development Courses/Certificate Mississippi Curriculum Framework

Apple/Swift

Same CIP as IST 11.0201 Computer Programming/ Programmer, General

Same CIP as IST 11.0202 Computer Programming, Specific Applications.

July 2019



Published by:

Mississippi Community College Board
Division of Workforce, Career, and Technical Education
3825 Ridgewood Road
Jackson, MS 39211
Phone: 601-432-6155
Email: curriculum@mccb.edu

FACULTY WRITING TEAM MEMBERS

Brandon Sesser, East Mississippi Community College
David Rose, Hinds Community College
Roderick Kwan, Hinds Community College
Kathy Boyte, Hinds Community College
Kenneth Boyte, Hinds Community College
Cody Robertson, Jones County Junior College
Robin Hayes, Mississippi Gulf Coast Community College
Dr. James Gruich, Mississippi Gulf Coast Community College
Natasha Lewis, Northeast Mississippi Community College
Nick Jackson, Northeast Mississippi Community College
Tony Bouthwell, Meridian Community College
Daniel Ethridge, Meridian Community College

ADMINISTRATOR WRITING TEAM MEMBERS

Joe Cook, Assistant Dean, East Mississippi Community College
Dr. Richie McAlister, Vice President, Meridian Community College
Lori Smith, Coordinator, Meridian Community College
Joseph Knight, Dean, Business Development, Meridian Community College
Sherry Franklin, Associate Vice President, Hinds Community College
Rod Tolbert, Dean, Jones County Junior College
Jason Mattox, Associate Vice President Northeast Mississippi Community College
John Shows, Associate Vice President, Mississippi Gulf Coast Community College
Dr. Erin Riggins, Dean, Mississippi Gulf Coast Community College

BUSINESS AND INDUSTRY CONTRIBUTING TEAM MEMBERS

William Stone, Stone CG, Booneville, MS

OFFICE OF CURRICULUM AND INSTRUCTION TEAM MEMBERS

Dr. Scott Kolle, Director of Curriculum and Instruction, Mississippi Community College Board
Dr. LaToya Sterling, Curriculum Specialist, Office of Curriculum and Instruction,
Mississippi Community College Board
Dr. Sheriece Robinson, Curriculum Specialist, Office of Curriculum and Instruction,
Mississippi Community College Board

The Office of Curriculum and Instruction (OCI) was founded in 2013 under the Division of Workforce, Career, and Technical Education at the Mississippi Community College Board (MCCB). The office is funded through a partnership with The Mississippi Department of Education (MDE), who serves as Mississippi's fiscal agent for state and federal Career and Technical Education (CTE) Funds. The OCI is tasked with developing statewide CTE curriculum, programming, and professional development designed to meet the local and statewide economic demand.

Copyright® 2019 by Mississippi Community College Board
For information, please contact curriculum@mccb.edu

Contents

| | |
|--|----|
| ADOPTION OF NATIONAL CERTIFICATION STANDARDS | 5 |
| INDUSTRY JOB PROJECTION DATA..... | 6 |
| ARTICULATION | 7 |
| TECHNICAL SKILLS ASSESSMENT | 7 |
| RESEARCH ABSTRACT | 8 |
| REVISION HISTORY:..... | 8 |
| PROGRAM DESCRIPTION..... | 9 |
| SUGGESTED COURSE SEQUENCE | 10 |
| 15-hour Technical Certificate | 10 |
| INFORMATION SYSTEM ELECTIVE COURSES | 11 |
| INFORMATION SYSTEM COURSES..... | 12 |
| IST 1283 Programming Principles with Swift..... | 12 |
| IST 2453 Mobile Application Development | 16 |
| Appendix A: RECOMMENDED TOOLS AND EQUIPMENT | 18 |
| Appendix B: Curriculum Definitions and Terms | 19 |
| Appendix C: Textbook List..... | 21 |

ADOPTION OF NATIONAL CERTIFICATION STANDARDS

Swift® Playgrounds is a revolutionary app for iPad that makes learning Swift® interactive and fun. It requires no coding knowledge, so it's perfect for students just starting out. Solve puzzles to master the basics using Swift® — a powerful programming language created by Apple® and used by the pros to build today's most popular apps. Then take on a series of challenges and step up to more advanced playgrounds designed by Apple® and other leading developers.

Announced in 2014, the Swift® programming language has quickly become one of the fastest growing languages in history. Swift® makes it easy to write software that is incredibly fast and safe by design. The goals for Swift® are ambitious: they want to make programming simple things easy, and difficult things possible.

For students, learning Swift® has been a great introduction to modern programming concepts and best practices. And because it is open, their Swift® skills will be able to be applied to an even broader range of platforms, from mobile devices to the desktop to the cloud.

www.Apple.com/Swift/playgrounds

www.Swift.org

INDUSTRY JOB PROJECTION DATA

The Software Developers occupations require a Bachelor's Degree. There is expected to be a 26.46% increase at the state level. Median annual income for this occupation is \$80,629.12 at the state level. A summary of occupational data from the State Workforce Investment Board Data Center is displayed below:

Table 1: Education Level

| Program Occupations | Education Level |
|---------------------------------------|-------------------|
| Software Developers, Applications | Bachelor's Degree |
| Software Developers, Systems Software | Bachelor's Degree |

Table 2: Occupational Overview

| | Region | State | United States |
|-----------------------------|-------------|-------------|---------------|
| 2016 Occupational Jobs | 4,581 | 4,581 | 1,233,163 |
| 2026 Occupational Jobs | 5,793 | 5,793 | 1,870,980 |
| Total Change | 1,212 | 1,212 | 637,817 |
| Total % Change | 26.46% | 26.46% | 51.72% |
| 2016 Median Hourly Earnings | \$38.76 | \$38.76 | \$49.75 |
| 2016 Median Annual Earnings | \$80,629.12 | \$80,629.12 | \$103,480.00 |
| Annual Openings | 121 | 121 | 63782 |

Table 3: Occupational Breakdown

| Description | 2016 Jobs | 2026 Jobs | Annual Openings | 2016 Hourly Earnings | 2016 Annual Earnings 2,080 Work Hours |
|---------------------------------------|-----------|-----------|-----------------|----------------------|--|
| Software Developers, Applications | 3,045 | 4,155 | 111 | \$40.57 | \$84,385.60 |
| Software Developers, Systems Software | 1,536 | 1,638 | 10 | \$36.70 | \$76,336.00 |
| Total | 4,581 | 5,793 | 121 | \$38.76 | \$80,620.80 |

Table 4: Occupational Change

| Description | Regional Change | Regional % Change | State % Change | National % Change |
|---------------------------------------|-----------------|-------------------|----------------|-------------------|
| Software Developers, Applications | 1,110 | 36.45% | 36.45% | 69.90% |
| Software Developers, Systems Software | 102 | 6.64% | 6.64% | 15.86% |

ARTICULATION

There is no secondary course that articulates to this program at this time.

TECHNICAL SKILLS ASSESSMENT

Colleges should report the following for students who complete the program with a career certificate, technical certificate, or an Associate of Applied Science Degrees for technical skills attainment. To use the approved Alternate Assessment for the following programs of study, colleges should provide a Letter of Notification to the Director of Career Technical Education at the MS Community College Board. Please see the following link for further instructions: <http://www.mccb.edu/wkfEdu/CTDefault.aspx>.

| CIP Code | Program of Study | |
|--------------------|-----------------------------|-----------------------------|
| 11.0201 11.0202 | App Development | |
| Level | Standard Assessment | Alternate Assessment |
| Career | | |
| Level | Standard Assessment | Alternate Assessment |
| Technical/AAS | Swift® Programming Language | |

RESEARCH ABSTRACT

In the spring of 2019, the office of Curriculum met with Apple Consultants and all Mississippi Community Colleges to determine the need for the apple swift courses. Two courses were developed for all students in the Information Systems Technology Programs of study to use as electives if desired. The Swift assessment can be taken for the purpose of certification in programming.

REVISION HISTORY:

2019 Mississippi Community College Board

PROGRAM DESCRIPTION

The Mobile App Development curriculum is designed to help build a solid foundation in programming fundamentals using Swift as the language. Throughout this course, students will get practical experience with the tools, techniques, and concepts needed to build a basic iOS app from scratch. Students will also learn user interface design principles, which are fundamental to programming and making great apps.

Swift is a powerful and intuitive programming language created by Apple for building apps. It makes programming easier, more flexible, and more fun. Swift is not only great for getting you started with coding. It is designed to scale from writing the simplest program, like “Hello, world!” to the world’s most advanced software.

SUGGESTED COURSE SEQUENCE

15-hour Technical Certificate

| Course Number | Course Name | Semester Credit Hours | SCH Breakdown | | Total Contact Hours | Contact Hour Breakdown | | Certification Information |
|---------------|---------------------------------------|-----------------------|---------------|-----|---------------------|------------------------|-----|---------------------------|
| | | | Lecture | Lab | | Lecture | Lab | Certification Name |
| IST 1283 | Programming Principles with Swift | 3 | 2 | 2 | 60 | | | |
| IST 1453 | *Principles of Mobile App Development | 3 | 2 | 2 | 60 | | | |
| IST 2453 | Mobile App Development | 3 | 2 | 2 | 60 | | | |
| | Technical Electives | 6 | 2 | 2 | 60 | | | |
| | Total | 15 | | | | | | |

***Course requiring swift certification**

INFORMATION SYSTEM ELECTIVE COURSES

| Course Number | Course Name | Semester Credit Hours |
|---------------|--|-----------------------------|
| IST 2533 | IT Project Management | 3 |
| IST 293 (1-3) | Special Problems in Information Systems Technology | 3 |
| | Instructor Approved Electives Per Local Community College | |
| | Total | |

INFORMATION SYSTEM COURSES

Course Number and Name: IST 1283 Programming Principles with Swift

Description: This course is designed to help students build a solid foundation in programming using Swift. Students get practical experience with the tools and techniques they'll need to build basic iOS apps from scratch. They'll also learn problem-solving skills and develop their own ways of breaking down problems into manageable pieces. As they gain more practice as programmers, they'll get better at designing clever and efficient solutions to coding challenges.

Hour Breakdown:

| Semester Credit Hours | Lecture | Lab | Contact Hours |
|-----------------------|---------|-----|---------------|
| 3 | 2 | 2 | 60 |

Prerequisite: Instructor approved

Student Learning Outcomes:

Students will

1. Learn how binary values work. Unit 1 Intro to Coding
2. Learn the iterative design process. Unit 1 Intro to Coding
3. Understanding of UI. Unit 1 Intro to Coding
4. Understanding of playground. Unit 1 Intro to Coding
5. Understanding of Swift programming. Unit 1 Intro to Coding
6. Understanding of the various layers of the Internet (i.e. IP, TCP, HTTP, and TLS). Unit 2 Functions and Abstractions
7. Learn how to create and interactive app prototype in Keynote. Unit 2 Functions and Abstractions
8. Learn how functions work. Unit 2 Functions and Abstractions
9. Learn about constants and variables. Unit 2 Functions and Abstractions
10. Understand encryption. Unit 3 Algorithms
11. Understand cryptography. Unit 4 Building Apps
12. Distinguish between symmetric and asymmetric encryption. Unit 4 Building Apps
13. Understand the importance of Certificate Authorities. Unit 4 Building Apps
14. Understand the importance of data source pattern and objects. Unit 4 Building Apps
15. Understand of actions and outlets. Unit 4 Building Apps

Apple Swift Units

Unit 1: Introduction to Coding (4 weeks)

- 1.1 Topic in Focus: How Computers Work
- 1.2 Explore Task Practice 1
- 1.3 Create Task Practice 1
- 1.4 Playground Basics
- 1.5 Naming and Identifiers
- 1.6 Strings
- 1.7 Hello, world!
- 1.8 First App

Unit 2: Functions and Abstraction (6 weeks)

- 2.2 Explore Task Practice 2
- 2.3 Create Task Practice 2
- 2.4 Functions
- 2.5 Boogie Bot
- 2.6 Constants and Variables

2.7 Types

2.8 Parameters and Results

Unit 3: Algorithms (6 weeks)

3.1 Topic in Focus: Privacy and Cybersecurity

3.2 Making Decisions

3.3 Instances, Methods, and Properties

3.4 Question Bot

3.5 Arrays and Loops

3.6 Defining Structures

Unity 4: Building Apps (10 weeks)

4.1 Topic in Focus: Cryptography

4.2 Explore Task (8 classroom hours)

4.3 Chat Bot

4.4 Actions and Outlets

4.5 Adaptive User Interfaces

4.6 Enumerations and Switch

4.7 Final Project

4.8 Create Task (12 classroom hours)

Textbook: AP Computer Science Principles with Swift – Apple iBook

Need:

- A Mac running macOS Mojave or High Sierra.
- Xcode 10, which can be downloaded from the Mac App Store.

Course Number and Name: IST 1453 Principles of Mobile App Development

Description: This course is designed to help students build a solid foundation in programming fundamentals using Swift as a language. Students get practical experience with the tools, techniques, and concepts needed to build a basic iOS app from scratch. They'll also learn user interface design principles, which is fundamental to programming and making great apps.

Hour Breakdown:

| Semester Credit Hours | Lecture | Lab | Contact Hours |
|-----------------------|---------|-----|---------------|
| 3 | 2 | 2 | 60 |

Prerequisite: Instructor approved

Student Learning Outcomes:

1. Understanding of Playgrounds. Lesson 1 Playground Basics
2. Learn Swift Programming. Lesson 2: Naming and Identifiers, Lesson 3: Strings, Lesson 8: Constants and Variables, Lesson 10: Parameters and Results, Lesson 11: Making Decisions, Lesson 12: Instances, Methods, Properties, Lesson 14 Arrays and Loops, Lesson 15 Defining Structures
3. Learn how to use Xcode to build and run an app. Lesson 4: Hello World, Lesson 5: First App, Lesson 7: Boogie Bot, Lesson 13 Question Bot
4. Learn how to define and call functions. Lesson 6: Functions
5. Evaluate Xcode software (Cocoa Frameworks/Objective-C/iOS simulator). Lesson 5: First App

Apple Swift Units

Lesson 1 Playground Basics

1. Type and change code in a playground
2. Where to look for results of your work
3. How to add notes that will help you remember what your code means.
4. What it looks like if something goes wrong.

Lesson 2: Naming and Identifiers

1. Declare a new constant using the let keyword.
2. Choose a meaningful, specific name for a value.
3. Refer to a previously declared value by its name

Lesson 3: Strings

1. Identify different types of characters.
2. Declare a new string constant using literal text, including emoji.
3. Combine two existing strings.
4. Combine a string of literal text with one or more previously declared values.
5. Use the Show Results button to view longer strings.
6. Combine a string with a number
7. Use escape characters to output special text.

Lesson 4: Hello World

1. Understand and describe the importance of logging
2. Add logging to code
3. Use the console for basic debugging

Lesson 5: First App

1. Start a new X Code project
2. Build and run an app in the iOS Simulator on a device
3. Show and hide inspectors in X Code.
4. Change the background color of a view.
5. Add a resource to an X code project
6. Edit the main storyboard in an X Code project.

Lesson 6: Functions

1. Describe why functions are an important concept.
2. Define and implement basic functions.

Lesson 7: Boogie Bot

1. Use functions to build complex routines out of smaller parts.
2. Describe how functions provided by other developers allow the student to build on their work.

Lesson 8: Constants and Variables

1. Understand and describe the difference between constants and variables.
2. Identify appropriate times to use variables over constants.

Lesson 9: Types

1. Understand and explain a variety of types from the Swift standard library.
2. Identify cases where the Swift types system will infer a symbol's type.

Lesson 10: Parameters and Results

1. Create a function that accepts and uses parameters
2. Create a function that returns a value.
3. Use appropriate naming convention's to clearly describe functions and parameters.

Lesson 11: Making Decisions

1. Compare values
2. Understand, explain, and use conditionals to write code that will run specific lines of code based on certain values.

Lesson 12: Instances, Methods, Properties

1. Create values without using literals.
2. Describe the difference between a type and an instance.
3. Describe the difference between a method and a property.
4. Use documentation to find out information about unfamiliar types.

Lesson 13 Question Bot

1. Move code written in a playground into the right place in X code.
2. Use conditionals to provide varied functionality based on user input.

Lesson 14 Arrays and Loops

1. Understand and explain common use cases for arrays.
2. Use arrays to manage collections of objects.
3. Iterate through an array to perform a common action on each item in the collection.

Lesson 15 Defining Structures

1. Understand and explain the importance of custom structures and their common use cases.
2. Design a custom structure to group related data into one type.
3. Define custom properties and methods for custom structures.

Course Number and Name: IST 2453 Mobile Application Development

Description: This course introduces students to programming technologies, design, and development related to apple mobile applications. Topics include accessing device capabilities, industry standards, operating systems, and programming for mobile applications using an iOS Software Development Kit (SDK).

Hour Breakdown:

| Semester Credit Hours | Lecture | Lab | Contact Hours |
|-----------------------|---------|-----|---------------|
| 3 | 2 | 2 | 60 |

Prerequisite: Instructor approved

Student Learning Outcomes:

1. Distinguish between the types of application platforms. Lesson 16: Question Bot 2, Lesson 17 Actions and Outlets
2. Recognize the benefits, implications, and significance of mobile applications Lesson 18 Adaptive, Lesson 19: Enumerations and Switch User Interfaces
3. Compare existing software technologies used in the development of mobile applications. Lesson 17 Actions and Outlets
4. Compare the hardware used by mobile applications. Lesson 18 Adaptive User Interfaces
5. Design effective user interfaces for mobile platforms. Lesson 18 Adaptive User Interfaces
6. Design and deliver a mobile application. Lesson 20 Final Project
7. Discuss the distribution methods of mobile applications. Lesson 17 Actions and Outlets, Lesson 20 Final Project
8. Identify and evaluate desktop applications. Lesson 17 Actions and Outlets
9. Identify and evaluate native mobile applications. Lesson 17 Actions and Outlets
10. Identify and evaluate Internet/Web-based applications. Lesson 17 Actions and Outlets
11. Discuss the convenience and mobility of mobile applications. Lesson 17 Actions and Outlets
12. Examine the competition among businesses and organizations caused by the emergence of mobile applications. Lesson 17 Actions and Outlets
13. Examine the security issues involved with mobile applications. Lesson 17 Actions and Outlets
14. Evaluate other software technologies as developed. Lesson 18 Adaptive User Interfaces
15. Investigate smartphone hardware and its capabilities. Lesson 17 Actions and Outlets
16. Investigate tablet hardware and its capabilities. Lesson 17 Actions and Outlets
17. Demonstrate effective use of screen real estate. Lesson 18 Adaptive User Interfaces
18. Demonstrate effective use of accelerometers, global positioning systems, and other onboard devices. Lesson 17 Actions and Outlets
19. Demonstrate effective use of touch input. Lesson 18 Adaptive User Interfaces
20. Demonstrate effective use of object-oriented programming in the design of mobile applications. Lesson 19: Enumerations and Switch User Interfaces, Lesson 20 Final Project
21. Compare the various channels for application distribution. Lesson 17 Actions and Outlets, Lesson 20 Final Project

Apple Swift Units

Lesson 16: Question Bot 2

1. Understand and explain the data source pattern.
2. Understand and explain data source objects and their relationship to other objects in an application
3. Implement a simple data source object that communicates data to and from the user interface layer of an application.

Lesson 17 Actions and Outlets

1. Explain actions and use them to run code when the user taps the user interface.
2. Explain outlets and use them to connect objects in the storyboard to code.

3. Understand, explain, and use a variety of UI Control events.

Lesson 18 Adaptive User Interfaces

1. Build a user interface that adapts to different screen sizes.
2. Display images in code.
3. Use code to play sounds.

Lesson 19: Enumerations and Switch

1. Understand and explain the benefits and proper use cases of enumerations.
2. Use a switch statement on an enum value to conditionally trigger specific code.

Lesson 20 Final Project

1. Integrate all of the concepts from the course into a basic Rock, Paper, Scissors app.
2. Integrate all of the concepts from the course into an app that displays text over photos.

Appendix A: RECOMMENDED TOOLS AND EQUIPMENT

CAPITALIZED ITEMS

Mac Book Pro 2.3 GHz dual core i5, 128 GB-Space (1per student)

NON-CAPITALIZED ITEMS

IPad Wi-Fi 32 GB (1 per student)

RECOMMENDED INSTRUCTIONAL AIDS

Apple TV 4K 32 GB

Appendix B: Curriculum Definitions and Terms

- Course Name – A common name that will be used by all community colleges in reporting students
- Course Abbreviation – A common abbreviation that will be used by all community and junior colleges in reporting students
- Classification – Courses may be classified as the following:
 - Career Certificate Required Course – A required course for all students completing a career certificate.
 - Technical Certificate Required Course – A required course for all students completing a technical certificate.
 - Technical Elective – Elective courses that are available for colleges to offer to students.
- Description – A short narrative that includes the major purpose(s) of the course
- Prerequisites – A listing of any courses that must be taken prior to or on enrollment in the course
- Corequisites – A listing of courses that may be taken while enrolled in the course
- Student Learning Outcomes – A listing of the student outcomes (major concepts and performances) that will enable students to demonstrate mastery of these competencies

The following guidelines were used in developing the program(s) in this document and should be considered in compiling and revising course syllabi and daily lesson plans at the local level:

- The content of the courses in this document reflects approximately 75% of the time allocated to each course. The remaining 25% of each course should be developed at the local district level and may reflect the following:
 - Additional competencies and objectives within the course related to topics not found in the state framework, including activities related to specific needs of industries in the community college district
 - Activities that develop a higher level of mastery on the existing competencies and suggested objectives
 - Activities and instruction related to new technologies and concepts that were not prevalent at the time the current framework was developed or revised
 - Activities that include integration of academic and career–technical skills and course work, school-to-work transition activities, and articulation of secondary and postsecondary career–technical programs
 - Individualized learning activities, including work-site learning activities, to better prepare individuals in the courses for their chosen occupational areas
- Sequencing of the course within a program is left to the discretion of the local college. Naturally, foundation courses related to topics such as safety, tool and equipment usage, and other fundamental skills should be taught first. Other courses related to specific skill areas and related academics, however, may be sequenced to take advantage of seasonal and climatic conditions, resources located outside of the school, and other factors. Programs that offer an Associate of Applied Science Degree must include all of the required Career Certificate courses, Technical Certificate courses **AND** a minimum of 15 semester hours of General Education Core Courses. The courses in the General Education Core may be spaced out over the entire length of the program so that students complete some academic and Career Technical courses each semester. Each community college specifies the actual courses that are required to meet the General Education Core Requirements for the Associate of Applied Science Degree at their college.

- In order to provide flexibility within the districts, individual courses within a framework may be customized by doing the following:
 - Adding new student learning outcomes to complement the existing competencies and suggested objectives in the program framework
 - Revising or extending the student learning outcomes
 - Adjusting the semester credit hours of a course to be up 1 hour or down 1 hour (after informing the Mississippi Community College Board [MCCB] of the change)

Appendix C: Textbook List

| Recommended Coding (Apple App Development) Text Book List | | |
|--|-------------------|-------------|
| CIP:11.0201 Apple | | |
| Book Title | Author (s) | ISBN |
| Everyone Can Code | Apple | eBook |
| Intro to App Development w/ Swift | Apple | eBook |
| App Development w/ Swift | Apple | eBook |
| AP Computer Science Principles with Swift | Apple | eBook |