Maryland CTE Programs of Study

Computer Programming III – Swift Capstone (2025)

GENERAL REQUIREMENTS. This course is recommended for students in Grades 11-12. 1

1 GENERAL REQUIREMENTS. This course is recommended for students in Grades 11-12. 1

INTRODUCTION 2

A Career and Technical Education (CTE) instruction provides content aligned with challenging academic standards and relevant technical knowledge and skills for students to further their education and succeed in current or emerging professions. 2.A

B The Digital Technology Career Cluster focuses on building linkages in IT occupations for entry level, technical, and professional careers related to the design, development, support, and management of hardware, software, multimedia, and systems integration services. 2.B

C The Computer Programming Program of Study provides high school students with comprehensive software development skills through programming languages. Students' progress from fundamental coding concepts to complex development projects while earning industry certifications and building professional portfolios. Graduates emerge prepared for both college computer science programs and immediate entry into software development careers, equipped with practical experience in current programming practices and software engineering principles. 2.C

D Computer Programming III – Swift Data Collections course builds on foundational Swift programming skills to develop sophisticated iOS applications with complex data management. Students will master higher-order collection functions, implement persistent data storage solutions, and create network-connected apps through hands-on projects. The curriculum explores protocol-oriented programming, advanced Swift techniques, and integration with core iOS frameworks. Students will design and develop a capstone iOS application incorporating these advanced concepts, demonstrating their ability to create professional- grade mobile software. Following Apple's Swift Data Collections curriculum, this course prepares students for advanced iOS development roles and provides the skills needed to create apps ready for App Store submission. 2.D

Skip to main content

E **Students will participate in at least two Career-Connected Education and Work-Based Learning experiences in this course, which might include informational interviews or job shadowing relevant to the program of study.** 2.E

F **Students are encouraged to participate in extended learning experiences through aligned Career and Technical Student Organizations (CTSOs). CTSOs are a cocurricular requirement in the Carl D. Perkins Act, and alignment to CTSO activities is an expectation for CTE programs in the state of Maryland.** 2.F

KNOWLEDGE AND SKILLS 3

A **The student demonstrates the necessary skills for career development, maintenance of employability, and successful completion of course outcomes. The student is expected to:** 3.A

1. Identify and demonstrate positive work behaviors that enhance employability and job advancement, such as regular attendance, promptness, proper attire, maintenance of a clean and safe work environment, and pride in work. 3.A.1
2. Demonstrate positive personal qualities such as flexibility, open-mindedness, initiative, active listening, and a willingness to learn. 3.A.2
3. Employ effective reading, writing, and technical documentation skills. 3.A.3
4. Solve problems using critical thinking techniques and structured troubleshooting methodologies. 3.A.4
5. Demonstrate leadership skills and collaborate effectively as a team member. 3.A.5
6. Implement safety procedures, including proper handling of hardware and following cybersecurity guidelines. 3.A.6
7. Exhibit an understanding of legal and ethical responsibilities in the IT field, following data privacy laws and best practices for security. 3.A.7
8. Demonstrate time-management skills and the ability to prioritize tasks in a technical setting. 3.A.8

B **The student identifies various career pathways in the computer programming field. The student is expected to:** 3.B

1. Develop a career plan that includes the necessary education, certifications, job skills, and experience for specific roles in computer programming. 3.B.1
2. Create a professional resume and portfolio that reflect skills, projects, certifications, and recommendations. 3.B.2
3. Demonstrate effective interview skills for roles in IT and computer programming. 3.B.3

C **The student develops technology and digital literacy skills. The student is expected to:** 3.C

1. Use technology as a tool for research, organization, communication, and problem-solving. 3.C.1
2. Use digital tools, including computers, mobile devices, collaboration platforms, and cloud services, to access, manage, and create information. 3.C.2

Skip to main content

3 Demonstrate proficiency in using emerging and industry-standard technologies, including virtualization tools, network management software, and cybersecurity applications. 3.C.3

4 Understand ethical and legal considerations for technology use, including the principles of data protection, copyright, and responsible technology use. 3.C.4

D The student integrates core academic skills into computer programming practices. The student is expected to: 3.D

1 Demonstrate the use of clear communication techniques, both written and verbal, that are consistent with industry standards. 3.D.1

2 Apply mathematical concepts such as Algebraic functions and equations, Boolean algebra and logic, basic statistics for data analysis, order of operations, number systems (binary, decimal), percentages and proportions, basic geometry (for graphics and spatial problems) and pattern recognition. 3.D.2

3 Use scientific principles, such as data collection, pattern recognition, systematic investigation, controlled testing, and documentation of process. 3.D.3

E The student can extend their Swift collection knowledge. The student is expected to: 3.E

1 Review and apply basic collection types (arrays, dictionaries, sets). 3.E.1

2 Create and manipulate nested collections for complex data structures. 3.E.2

3 Implement custom collection initialization techniques. 3.E.3

4 Compare and select appropriate collection types for specific scenarios. 3.E.4

5 Apply appropriate access patterns to retrieve data efficiently. 3.E.5

F The student can implement higher-order collection functions. The student is expected to: 3.F

1 Use the map function to transform collection elements. 3.F.1

2 Apply filter to create subsets of collections based on criteria. 3.F.2

3 Implement reduce to combine collection elements into a single value. 3.F.3

4 Create custom sorting with closure expressions. 3.F.4

5 Chain multiple higher-order functions for complex data transformations. 3.F.5

G The student can implement persistent data storage. The student is expected to: 3.G

1 Save and load basic data using the file system. 3.G.1

2 Store user preferences using User Defaults. 3.G.2

3 Implement data serialization using Codable protocol. 3.G.3

4 Handle data persistence errors appropriately. 3.G.4

5 Design data models that support effective serialization. 3.G.5

Skip to main content

H **The student can design and implement protocols and extensions. The student is expected to:** 3.H

1. Create custom protocols to define shared behavior. 3.H.1
2. Extend existing types with new functionality. 3.H.2
3. Implement protocol conformance for custom types. 3.H.3
4. Apply protocol-oriented design principles to solve problems. 3.H.4
5. Use protocol extensions to provide default implementations. 3.H.5

I **The student can perform basic networking operations. The student is expected to:** 3.I

1. Create network requests using URL Session. 3.I.1
2. Handle network responses and errors appropriately. 3.I.2
3. Parse JSON data from web services. 3.I.3
4. Implement asynchronous data fetching. 3.I.4
5. Update user interfaces based on network results. 3.I.5

J **The student can create advanced user interface components. The student is expected to:** 3.J

1. Implement custom table and collection views. 3.J.1
2. Create custom cells with dynamic content. 3.J.2
3. Handle user interaction with complex UI components. 3.J.3
4. Apply proper data source and delegate patterns. 3.J.4
5. Implement search and filtering in collection-based interfaces. 3.J.5

K **The student can build advanced Swift interfaces. The student is expected to:** 3.K

1. Create custom view components for reusability. 3.K.1
2. Implement state management with property wrappers. 3.K.2
3. Design responsive layouts that adapt to different screen sizes. 3.K.3
4. Apply animations and transitions between view states. 3.K.4
5. Share data across views using environment objects. 3.K.5

L **The student can integrate system frameworks. The student is expected to:** 3.L

1. Implement basic location services using Core Location. 3.L.1
2. Display maps and annotations with MapKit. 3.L.2
3. Create and schedule local notifications. 3.L.3
4. Access and use the device camera and photo library. 3.L.4
5. Implement basic device authentication features. 3.L.5

Skip to main content

M **The student can apply app architecture patterns. The student is expected to:** 3.M

1. Organize code using the Model-View-Controller pattern. 3.M.1
2. Implement proper separation of concerns in app design. 3.M.2
3. Apply the delegate pattern for component communication. 3.M.3
4. Create reusable components with clearly defined responsibilities. 3.M.4
5. Design data flow strategies for complex apps. 3.M.5

N The student can perform testing and debugging. The student is expected to: 3.N

1. Write basic unit tests for Swift code. 3.N.1
2. Debug common issues using Xcode's debugging tools. 3.N.2
3. Implement logging for troubleshooting. 3.N.3
4. Test app functionality across different scenarios. 3.N.4
5. Create strategies for handling potential runtime errors. 3.N.5

O The student can develop a capstone iOS application. The student is expected to: 3.O

1. Plan and design an app that solves a specific problem. 3.O.1
2. Create wireframes and prototypes for the user interface. 3.O.2
3. Implement core functionality using appropriate Swift features. 3.O.3
4. Test the app thoroughly on multiple device sizes. 3.O.4
5. Present and explain the technical implementation details. 3.O.5