# Software Development

Candidates for this exam are seeking to prove core software development skills, including object-oriented programming, web applications, and databases. Candidates are expected to have some experience with C# and ANSI SQL.

Candidates should have at least 150 hours of instruction or hands-on experience with concepts related to programming, software development, object-oriented programming, web applications, and databases.

To be successful on the test, the candidate is also expected to have the following prerequisite knowledge and skills:

- HTML
- CSS
- JavaScript
- At least one object-oriented language (C#, Java, C++)
- SQL queries, XML, JSON

## 1. Core Programming Concepts

### 1.1 Describe computer storage and data types

- How a computer stores programs and instructions in computer memory, memory stacks and heaps, memory size requirements for various data storage types, numeric data and textual data, garbage collection

### 1.2 Construct and analyze algorithms and flowcharts to solve programming problems

- Decision structures used in all computer programming languages; if decision structures; multiple decision structures, such as if...else and switch; reading and constructing flowcharts; decision tables; evaluating expressions;  for loops, while loops, do...while loops; recursion

### 1.3 Incorporate error handling into applications or modules

- Structured exception handling (try-catch-finally), unit testing, throwing exceptions, reading the stack, defensive coding, understand scoping in exception handling, writing to the console, breakpoints, step-into, step-over

### 1.4 Construct and analyze code based on functional programming patterns

- Event, delegate, promises, synchronous vs. asynchronous programming (AJAX, XHR), immutability

## 2. Software Development Principles

### 2.1 Describe software development lifecycle (SDLC) management

- Requirement analysis, planning and design, implementation, testing, deployment, maintenance, Agile concepts (SCRUM, sprints, product owner, stakeholders); types of testing (unit, integration, system, user acceptance, smoke testing, performance testing, load testing)



© 2025 Certiport, Inc. Certiport and the Certiport logo are registered trademarks of Certiport Inc. All other trademarks and registered trademarks are the property of their respective holders.

### 2.2 Interpret application specifications

- Reading application specifications and translating them into prototypes and code, selecting appropriate application type and components, using Generative AI to research the best approach

### 2.3 Construct and analyze code that uses algorithms and data structures

- Arrays, stacks, queues, linked lists, dictionaries (key-value pairs), sorting algorithms (selection sort, bubble sort, quick sort, merge sort), searching algorithms (linear search, binary search), performance implications of various data structures, choosing the right data structure, FIFO, LIFO

### 2.4 Describe the purpose of version control systems

- GitHub, check-in, check-out, merge, branch, rollback, clone, resolving conflict, code review

### 2.5 Describe secure coding concepts

- Encryption, hashing, and digital signatures; public, private, and shared keys; mitigating cross-site request forgery (csrf); SQL injection; risks of using iframes

## 3. Object-Oriented Programming

### 3.1 Construct, analyze, and use classes

- Properties, methods, events, fields, and constructors; how to create classes; how to use classes in code; access modifiers; instantiation; static vs. instance; encapsulation; composition

### 3.2 Construct and analyze code that uses inheritance

- Inherit the functionality of a base class into a derived class, generic classes, abstract classes

### 3.3 Construct and analyze code that uses polymorphism

- Extending the functionality of a class after inheriting from a base class, overriding methods in the derived class, interfaces, overloading

## 4. Web Applications

### 4.1 Construct and analyze web applications

- HTML5, CSS3, and JavaScript ES6; browser developer tools; HTTP request or response; state management; cookies, local, and session storage; page lifecycle; event model; client-side vs. server-side programming

### 4.2 Describe and configure web hosting

- Creating virtual directories and websites, publishing web applications, role of the web server

### 4.3 Describe and configure web services

- Web services that are consumed by client applications, accessing web services from client applications, JSON, REST API, oAuth, XML

### 4.4 Describe and identify architectural patterns

- Model-view-controller (MVC), model-view-viewmodel (MVVM), single page application (SPA)



© 2025 Certiport, Inc. Certiport and the Certiport logo are registered trademarks of Certiport Inc. All other trademarks and registered trademarks are the property of their respective holders.

## 5. Databases

### 5.1 Design and normalize a database

- Characteristics and capabilities of database products, database design, Entity Relationship Diagrams (ERDs), normalization concepts (to 3NF), indexes, constraints, primary key, foreign keys, relationships, cardinality

### 5.2 Construct, analyze, and optimize ANSI SQL queries

- Creating and accessing stored procedures, updating and selecting data, DML vs. DDL, functions, triggers, cursors, joins, indexes

### 5.3 Manage transactions

- Commit, rollback, save, concurrency, isolation levels, lock

### 5.4 Describe database access methods

- Entity Framework (Code-first, Database-first), connection pools, LINQ

### 5.5 Describe types of NoSQL databases

- Document databases, key-value databases



© 2025 Certiport, Inc. Certiport and the Certiport logo are registered trademarks of Certiport Inc. All other trademarks and registered trademarks are the property of their respective holders.