Software Development

Employer Signaling System by the Greater Washington Partnership

Generated: February 18, 2026

Database, Networking, Distributed System Fundamentals

Label	KSAC Description	KSA	Bloom's Taxonomy Level
a	Ability to normalize a database through 3rd normal form.	ability	3
b	Define various types of databases	knowledge	3
c	Ability to perform database schema design.	ability	3
d	Apply SQL data manipulation language such as Select (From), Insert, Update, Delete, JOIN (inner, outer, full, left, right), Where, Group By, Order By, etc.	ability	3
e	Knowledge of HTTP, HTTPS, SSL.	knowledge	3
f	Describe Infrastructure as code.	knowledge	2
g	Demonstrate an understanding of cloud architecture and the capabilities of services such as AWS, Azure, IBM, Oracle and Google.	knowledge	2
h	Ability to install and configure software.	ability	3
i	Knowledge of computer hardware how compute resources, storage, and networking interact in cloud and on-prem environments.	knowledge	2
l	Explain data security in terms of authentication, authorization, access and auditing.	knowledge	3
n	Identify and apply modern transport patterns, secure communication, and API calls.	skill	2
o	Identify and differentiate structured vs unstructured data.	knowledge	2
p	Explain how modern distributed systems (microservices, event-driven systems) operate.	knowledge	1
q	Explain API communication patterns (REST, GraphQL, gRPC).	knowledge	1

Label	KSAC Description	KSA	Bloom's Taxonomy Level
r	Describe cloud-native application design (functions-as-a-service, containers, serverless).	knowledge	1
s	Explain tradeoffs between relational, NoSQL, and vector databases.	knowledge	1
t	Describe system observability concepts (metrics, logs, traces).	knowledge	1
u	Explain data privacy requirements (PII handling, data minimization).	knowledge	1

Fundamental Programming Concepts

Label	KSAC Description	KSA	Bloom's Taxonomy Level
a	Describe uses of primitive data types.	knowledge	1
b	Write and reason about code using appropriate data types for performance, readability, and maintainability.	skill	3
c	Demonstrate fundamental programming skills including the use of variables, loops, conditional branching, and program logic, with an emphasis on error-handling and input validation.	skill	3
d	Design, implement, test, and debug a program that uses each of the following fundamental programming constructs: basic computation, simple I/O, standard conditional and iterative structures.	skill	3
f	Design and implement a program that defines functions and passes parameters.	skill	3
g	Write a program that uses file I/O to provide persistence across multiple executions.	skill	2
h	Choose appropriate conditional and iteration constructs for a given programming task.	skill	4
j	Describe the concept of recursion and give examples of its use.	knowledge	1

Label	KSAC Description	KSA	Bloom's Taxonomy Level
k	Valdiate and test AI-generated code for correctness, security, and reliability.	ability	1
l	Debug unfamiliar codebases using modern tools.	ability	1
m	Use version control branching strategies (feature branches, pull requests).	ability	1
n	Explain asynchronous programming models (async/await, event loops).	knowledge	1
o	Understand memory models relevant to modern programming languages (managed vs unmanaged).	knowledge	1

Fundamental Data Structures

Label	KSAC Description	KSA	Bloom's Taxonomy Level
a	Apply built-in libraries and interfaces to implement data structures in real codebases.	ability	1
b	Describe common applications for data structures such as: stack, queue, priority queue, set, and map.	knowledge	1
c	Write programs that use each of the following data structures: arrays, records/structs, strings, linked lists, stacks, queues, sets, and maps.	skill	3
d	Write programs that can use each of the following data structures: heaps and B-trees.	skill	3
e	Compare alternative implementations of data structures with respect to performance.	skill	4
f	Choose the appropriate data structure for modeling a given problem.	skill	3
g	Identify the costs and benefits of dynamic and static data structure implementations.	knowledge	1
h	Understand fundamental data management and storage principles.	knowledge	1

Label	KSAC Description	KSA	Bloom's Taxonomy Level
i	Explain time and space complexity tradeoffs for choosing a data structure.	knowledge	1
j	Understand immutability and persistent data structures in functional programming contexts.	knowledge	1

Development Methods

Label	KSAC Description	KSA	Bloom's Taxonomy Level
a	Trace the execution of a variety of code segments and write summaries of their computations.	knowledge	2
b	Explain why the creation of program components with minimal defects is important in the production of high-quality software.	knowledge	1
c	Describe the factors that influence performance in a production environment (load, resource intensive tasks, dependencies on external interfaces).	knowledge	2
d	Conduct a personal code review (focused on common coding errors) on a program component using a provided checklist.	skill	3
e	Contribute to a small-team code review focused on component correctness.	skill	3
f	Describe how a contract can be used to specify the behavior of a program component.	knowledge	2
g	Refactor a program by identifying opportunities to apply procedural abstraction.	skill	2
h	Apply a variety of strategies to the testing and debugging of simple programs.	ability	4
i	Construct and debug programs using the standard libraries available within a chosen programming language.	ability	4
j	Analyze the extent to which another programmer’s code meets documentation and programming style standards.	ability	4

Label	KSAC Description	KSA	Bloom's Taxonomy Level
k	Apply consistent documentation and program style standards that contribute to the readability and maintainability of software.	skill	3

Algorithms and Design

Label	KSAC Description	KSA	Bloom's Taxonomy Level
a	Explain the importance of algorithms in the problem-solving process.	knowledge	2
b	Explain how a problem may be solved by multiple algorithms, each with different properties.	knowledge	2
c	Create algorithms for solving simple problems.	skill	3
d	Use a programming language to implement, test, and debug algorithms for solving simple problems.	skill	3
e	Implement, test, and debug simple recursive functions and procedures.	skill	3
f	Determine whether a recursive or iterative solution is most appropriate for a problem.	skill	3
g	Implement a divide-and-conquer algorithm for solving a problem.	skill	3
h	Apply the techniques of decomposition to break a program into smaller pieces.	skill	3
i	Implement a coherent abstract data type, with loose coupling between components and behaviors.	skill	3
j	Evaluate the relative strengths and weaknesses among multiple designs or implementations for a problem.	skill	4

Software Design

Label	KSAC Description	KSA	Bloom's Taxonomy Level

a	Articulate design principles including separation of concerns, information hiding, coupling and cohesion, and encapsulation.	knowledge	2
b	Use a design paradigm to design a small-scale application or service, and explain how system design principles have been applied in this design.	skill	2
c	Construct models of the design of a small-scale application or service that are appropriate for the paradigm used to design it.	ability	2
d	Describe one or more design patterns that could be applicable to the design of a simple software within the context of a single design paradigm.	knowledge	2
f	Discuss and select an appropriate design paradigm for a simple system for a given scenario.	skill	3
g	Create appropriate models for the structure and behavior of software products from their requirements specifications.	ability	4
h	Explain the relationships between the requirements for a software product and its design, using appropriate models.	ability	3
i	Describe the software architecture for the design of a simple software system.	knowledge	3
j	Identify an appropriate software architecture by differentiating among common software architectures such as 3-tier, pipe-and-filter, message-queue, and client-server for a high-level design.	skill	3
k	Investigate the impact of software architecture selection on the design of a simple system.	skill	4
l	Apply simple examples of patterns in software design.	skill	3
m	Describe a form of refactoring and discuss when it may be applicable.	knowledge	2
n	Select suitable components for use in the design of a software product.	ability	3
o	Describe pros and cons of building a new component vs adapting an existing component (e.g. open source) or buying a component in the design of a software product.	knowledge	3
p	Explain the role of objects in middleware systems and the relationship with components.	skill	4

q	Refactor code for performance, readability, modularity, or security.	ability	4
r	Understand the user journey and the role it plays in the software development process.	knowledge	2
s	Understand when to use low code or no code approaches in the software development process.	knowledge	2
l	Evaluate architectural tradeoffs (latency, cost, scalability).	knowledge	1
m	Explain modern architecture patterns (microservices, event-driven, serverless).	knowledge	1
n	Understand how privacy, security, and governance requirements shape architecture.	knowledge	1

Software Construction and Analytics

Label	KSAC Description	KSA	Bloom's Taxonomy Level
a	Identify and evaluate the key components, including data privacy and ethical considerations, of a use-case or similar description of some behavior that is required for a system using the requirements-gathering process.	knowledge	2
b	Describe the requirements engineering process to elicit and validate behavioral requirements.	knowledge	2
c	Interpret a use-case or similar requirements model for a simple software system.	skill	2
d	Identify both functional and non-functional requirements in a given requirements specification for a software system.	skill	2
e	Apply key elements and common methods for elicitation and analysis to produce a set of software requirements for a small-sized software system.	skill	3
f	Describe the relative advantages and disadvantages among several major process models (e.g., waterfall, iterative, and agile).	knowledge	1
g	Describe the different practices that are key components of various process models.	knowledge	2

Label	KSAC Description	KSA	Bloom's Taxonomy Level
h	Differentiate among the phases of software development and understand associated roles and responsibilities of teammates for each.	knowledge	2
i	Ability to executive phases of software development in iterative and agile manner.	ability	3
j	Understand how programming in the large differs from individual efforts including a large code base, code reading, builds, context and the documentation of changes.	knowledge	2
k	Ability to participate in an agile software development team.	ability	3
l	Ability to work on a team; responsibilities of how each role and how they fit together.	ability	3
m	Knowledge of the Agile manifesto.	knowledge	2
n	Understanding the meaning of minimum viable product.	knowledge	2
o	Knowledge of DevOps.	knowledge	2
p	Ability to contextualize, modernize, and optimize both legacy and emerging software systems.	ability	2
q	Apply reverse engineering skills to understand unfamiliar software.	ability	2
r	Understand product metrics, analytics instrumentation, and event logging.	knowledge	1
s	Understand MVP thinking in agile product delivery.	knowledge	1
t	Apply requirements in iterative cycles rather than linear processes.	ability	2

Software Tools and Environment

Label	KSAC Description	KSA	Bloom's Taxonomy Level

a	Describe the issues that are important in selecting a set of tools for the development of a particular software system, including tools for requirements tracking, design modeling, implementation, build automation, and version control.	knowledge	2
b	Construct, execute and debug programs using a modern IDE and associated tools such as unit testing tools and visual debuggers.	ability	4
c	Describe how version control can be used to help manage software release management.	knowledge	3
d	Ability to use Github or similar version control systems.	ability	2
e	Ability to use Jira or other similar issue tracking tools.	ability	2
f	Ability to use npm, maven or similar package management tools (e.g. npm, maven, or similar).	ability	2
g	Ability to use a source code control tool to manage configuration items in a small team-based project.	ability	3
h	Explain the use of software integration tools.	knowledge	2
i	Describe the benefits of integrating static and dynamic test tools into the software development environment.	knowledge	2
j	Demonstrate the capability to use software tools in support of the development of a software product of small size.	ability	3
k	Explain the benefits of and appropriate uses cases for the usage of containers.	knowledge	2
m	Use AI-assisted development tools while validating results.	ability	1
n	Apply CI/CD pipelines in practice, not just identify CI tools.	ability	2
o	Use containerization for development, testing, and deployment workflows.	skill	2
p	Understand modern package management ecosystems (pip, pnpm, Go modules).	knowledge	1

Software Verification and Validation

Label	KSAC Description	KSA	Bloom's Taxonomy Level

a	Distinguish among the different types and levels of testing (unit, integration, systems, and acceptance).	knowledge	1
b	Describe techniques for identifying significant test cases for integration, regression and system testing.	knowledge	2
c	Create and document a set of tests for a medium-size code segment.	skill	3
d	Describe how to recognize, create, and automate continuous regression tests.	knowledge	3
e	Conduct collaborative inspections or reviews of software source code for common pitfalls or application of coding standards on a medium sized software project.	ability	3
f	Understanding of quality assurance processes, tools, and quality assurance team role in development.	knowledge	2
g	Build robust code using exception handling mechanisms.	skill	3
h	Describe secure coding and defensive coding practices.	knowledge	2
i	Rewrite a simple program to remove common vulnerabilities, such as buffer overflows, integer overflows and race conditions.	ability	3
j	Apply a defined coding standard in a small software project.	skill	3
k	Adhere to the defined coding standard in the project for new code.	skill	2
l	Write a software component that performs some non-trivial task and is resilient to common input and run-time errors.	ability	5
m	Be able to give or receive effective peer code reviews of a single commit or function.	skill	3
n	Inspect, research, and review open source code to identify privacy and security concerns.	ability	3
o	Monitor, analyze, and verify the output of generative models.	ability	3
p	Write automated tests using unit, integration, and end-to-end frameworks.	skill	2
q	Use mocking, stubbing, and dependency injection to isolate test cases.	ability	2

r	Apply static analysis and dynamic analysis tools in the development lifecycle.	ability	2

Secure Coding Principles

Label	KSAC Description	KSA	Bloom's Taxonomy Level
a	Identify common coding errors that lead to insecure programs (e.g., buffer overflows, memory leaks, malicious code) and apply strategies for avoiding such errors.	skill	3
b	Knowledge of principles of cybersecurity.	knowledge	2
c	Understanding of secure coding principles.	knowledge	3
d	Apply the principles of least privilege, defensive programming, and fail-safe defaults.	ability	3
e	Ability to determine what information can and more importantly can not be logged (e.g. compliance considerations).	knowledge	2
f	Write code with logging capabilities.	skill	2
g	Awareness of Defensive programming.	knowledge	1
h	Understand basics of securing web apps - SQL Injection and other input validation (parameterized queries, ORM safety, and input sanitization).	knowledge	3
i	Ability to determine the origins of open source code.	ability	3
j	Understand software bill of materials (SBOM).	knowledge	2
k	Understanding and contextualizing sectoral differences between developing for public sector and private sector.	knowledge	2
l	Understand supply-chain security (dependency risks, package integrity).	knowledge	1
m	Understand secure API design, authentication flows (OAuth2, JWT), and session handling.	knowledge	1