Exam 98-382: Introduction to Programming Using JavaScript – Skills Measured

Audience Profile

Candidates for this exam should be able to recognize and write syntactically correct JavaScript code, use data types supported by JavaScript, and be able to recognize and write JavaScript code that will logically solve a given problem.

Candidates are expected to have at least 100 hours of instruction or hands-on experience with the JavaScript programming language. Candidates should also be familiar with JavaScript features and capabilities, and understand how to write, debug, and maintain well-formed, well documented JavaScript code.

Skills Measured

NOTE: The bullets that appear below each of the skills measured are intended to illustrate how we are assessing that skill. This list is not definitive or exhaustive.

NOTE: In most cases, exams do NOT cover preview features, and some features will only be added to an exam when they are GA (General Availability).

Program with JavaScript Operators, Methods, and Keywords (20-25%)

Complete or debug code that uses assignment and arithmetic operators

- assignment; increment; decrement; addition; subtraction; division; multiplication; modulus; compound assignment operators

Apply JavaScript best practices

- comments; indentations; naming conventions; noscript; constants; reserved keywords; debugger keyword; setting breakpoints; console.log

Evaluate the use of inline and external scripts

- when to use, how to use, and what happens when both are used

Implement exception handling

- try; catch; finally

Complete and debug code that interacts with the Browser Object Model (BOM)

- manage state; display dialogs; determine screen size

Program with Variables, Data Types, and Functions (25-30%)

Declare and use variables of primitive data types

- number; boolean; string; null; undefined; typeof operator; type checking functions; use strict; converting between data types; formatting numbers; string operations; single quote vs double quote (nesting); initialization

Declare and use arrays

- single-dimensional arrays; multi-dimensional arrays; iteration; initialization; define an array; sort and search an array; use push, pop, shift, and unshift methods; use the length property; access an array element

Complete and debug code that uses objects

- properties; methods; instantiation; date object; retrieve date and time parts; localize date format (MM/DD vs DD/MM); add and subtract dates

Complete and debug code that uses built-in Math functions

- random; round; abs; floor; ceiling; min; max; pow; sqrt

Complete and debug a function that accepts parameters and returns a value

- reusable code; local versus global scope, redefine variables, pass parameters, value versus reference, return values

Implement and Analyze Decisions and Loops (20-25%)

Evaluate expressions that use logical and comparison operators

- ==; !=; <, >; <=; >=; !; &&; ||

Complete and debug decision statements

- if; else if; switch; nested if

Complete and debug loops

- for; while; do; break; continue

Interact with the Document Object Model (15-20%)

Identify and construct the Document Object Model (DOM) tree

- window; document; body; other HTML elements

Identify and handle HTML events

- onchange; onmouseover; onload; onclick; onmouseout; onkeydown

Complete and debug code that outputs to an HTML document

- innerHTML; document.write

Complete and debug code that locates, modifies, and adds HTML elements and attributes

- getElementById; getElementsByTagName; getElementsByClassName; setAttribute; createElement

Interact with HTML Forms (5-10%)

Complete and debug code that retrieves input from forms and sets form field values

- retrieve form values; identify the DOM path; get values from different types of elements; prepopulate values; mask values

Complete and debug code that performs input validation

- case; string comparisons; Not-A-Number (NaN)

Describe the form submission process

- onsubmit; post versus get; potential targets for submission