# WAS Exam Content Outline

**I. Creating Accessible Web Solutions (40%).**

1. Guidelines, principles and techniques for meeting success criteria (including WCAG 2.1, WAI-ARIA, ATAG, basic concepts, limitations of the specific guidelines, principles, and techniques, what is normative vs. non-normative; what is included in different levels (A, AA, AAA)).
2. Basic knowledge of programming (at a conceptual level; principles and concepts related to programming; impact of specific coding practices on web solutions vs. writing specific code).
3. Accessibility quality assurance (i.e., assuring the quality of accessibility throughout the development life cycle, difference and overlap between user experience and accessibility).
4. Accessibility supported technologies (including user's assistive technologies and accessibility features; combination of assistive technologies and users agent; design decisions in choosing technologies that support accessibility; e.g., not choosing Flash when something else has better accessibility support, differences in assistive technology supports and behaviors, differences in support for touch when screen reader is on vs. off).
5. Standard controls vs. custom controls (e.g., using standard controls when possible; if using custom controls build them using WAI-ARIA best practices).
6. Single page applications (e.g., focus control, delays for AJAX-Screen Reader compatibility, live announcements).
7. Strategies of persons with disabilities in using web solutions (e.g., navigation of screen reader users, headings and landmarks, coping strategies, user preferred methods vs. website specific methods, using keyboard vs. mouse).

**II. Identify accessibility issues in web solutions (40%).**

1. Interoperability and compatibility issues.
2. Identifying guidelines and principles regarding issues (including WCAG 2.1, WAI-ARIA, ATAG, basic concepts, limitations of the specific guidelines, principles, and techniques, what is normative vs. non-normative; what is included in different levels (A, AA, AAA)).
3. Testing with assistive technologies (e.g., navigation of screen reader users, headings and landmarks, screen magnifiers, high contrast, using keyboard vs. mouse).
4. Testing for end-user impact (e.g., low vision, cognitive, mobile/touch).
5. Testing tools for the web (both automated and manual tools, i.e., what they are and what are their limitations; e.g., unit testing, browser based tools, spider tools, bookmarklet, automated tools used to monitor site vs. external tools).

**III. Remediating issues in web solutions (20%).**

1. Level of severity and prioritization of issues (e.g., cost benefit; legal risk, user impact, what is the problem, what to focus on first).
2. Recommending strategies and/or techniques for fixing issues (i.e., best solution, solution that most widely useful, feasibility of solution, fixing vs. redesign, how to fix it).