

70-481: Essentials of developing Windows Store apps using HTML5 and JavaScript

The following tables itemize changes to exam 70-481. These changes will be made on November 18, 2013, to include updates that relate to Windows 8.1 and Visual Studio 2013 tasks.

1. Design Windows Store apps (no change: 20%)

Tasks currently measured	Tasks to be added/changed in November 2013
<p>Design the UI layout and structure Evaluate the conceptual design; decide how the UI will be composed; design for the inheritance and re-use of visual elements (e.g., styles, resources); design for accessibility; decide when custom controls are needed</p>	<p>Added sub-task: use the Hub App template</p>
<p>Design for separation of concerns Plan the logical layers of your solution to meet application requirements; design loosely coupled layers; incorporate WinMD components</p>	<p>No change</p>
<p>Design and implement Process Lifetime Management (PLM) Choose a state management strategy; handle the suspend event (oncheckpoint); prepare for app termination; handle the onactivated event; check the ActivationKind and previous state</p>	<p>No change</p>
<p>Plan for an application deployment Plan a deployment based on Windows 8 Application certification requirements; prepare an app manifest (capabilities and declarations); sign an app; plan the requirements for an enterprise deployment</p>	<p>No change</p>

2. Develop Windows Store apps (decrease: 20% to 19%)

Tasks currently measured	Tasks to be added/changed in November 2013
<p>Access and display contacts Call the ContactsPicker (windows.applicationmodel.contacts) class; filter which contacts to display; display a set number of contacts; create and modify contact information; select specific contact data</p>	<p>No change</p>
<p>Design for charms and contracts Choose the appropriate charms based on app requirements; design an application to be charm- and contract-aware; configure the application manifest for correct permissions</p>	<p>No change</p>
<p>Implement search Provide search suggestions using the SearchPane class; search and launch other apps; provide and constrain search within an app, including inside and outside of search charm; provide search result previews; implement activation from within search</p>	<p>Modified subtask: Provide search suggestions using the SearchPane class and SearchBox control</p> <p>New subtask: configure search contracts</p>
<p>Implement Share in an app Use the DataTransferManager class to share data with other apps; accept sharing requests by implementing activation from within Share; limit the scope of sharing using the DataPackage object; implement in-app Share outside of Share charm</p>	<p>New subtask: use web links and application links</p>
<p>Manage application settings and preferences Choose which application features are accessed in AppSettings; add entry points for AppSettings in the Settings window; create settings flyouts using the SettingsFlyout control; add settings options to SettingsFlyout; store and retrieve settings from the roaming app data store</p>	<p>No change</p>
<p>None</p>	<p>New task (section) Integrate media features Support DDS images; implement video playback; implement XVP and DXVA; implement TTS; implement audio and video playback using HTML5 DRM</p>

3. Create the user interface (decrease: 21% to 20%)

Tasks currently measured	Tasks to be added/changed in November 2013
<p>Implement WinJS controls Flipview; flyout; grid layout; list layout; menu object</p>	<p>New subtasks: WebView; item container; repeater</p>
<p>Implement HTML layout controls Implement layout controls to structure your layout; support scrolling and zooming with CSS3; manage text flow and presentation, including overflow</p>	<p>New subtask: implement templates and bindings</p>
<p>Create layout-aware apps to handle view states Handle ViewState events; choose among application view states; use CSS3 media queries to handle different view states; respond to ApplicationViewStateChangeEvent</p>	<p>Revised task: Create layout-aware apps to handle windowing modes</p> <p>Revised, complete list of subtasks: use CSS3 media queries to adapt to different devices; respond to changes in orientation; adapt to new windowing modes by using the ViewManagement namespace; manage settings for an apps view</p>
<p>Design and implement the app bar Determine what to put on the app bar based on app requirements; style and position app bar items; design the placement of controls on the app bar; handle AppBar events</p>	<p>No change</p>
<p>Apply CSS styling Implement gradients, grid layouts, zooming, scroll snapping, and media queries</p>	<p>No change</p>

4. Program user interaction (no change: 20%)

Tasks currently measured	Tasks to be added/changed in November 2013
Manage input devices Capture gesture library events; create custom gesture recognizers; listen to mouse events or touch gestures; manage Stylus input and inking; handle drag and drop events	No change
Design and implement navigation in an app Handle navigation events, check navigation properties, and call navigation functions by using the WinJS.Navigation namespace; design navigation to meet app requirements; Semantic Zoom; load HTML fragments	No change
Create and manage tiles Create and update tiles and tile contents; create and update badges (the TileUpdateManager class); respond to notification requests; choose an appropriate tile update schedule based on app requirements	No change
Notify users by using toast Enable an app for toast notifications; populate toast notifications with images and text by using the ToastUpdateManager; play sounds with toast notifications; respond to toast events; control toast duration	New subtask: configure and use Azure Mobile Services for push notifications

5. Manage security and data (increase: 19% to 21%)

Tasks currently measured	Tasks to be added/changed in November 2013
<p>Choose a data access strategy Choose the appropriate data access strategy (file based; web service; remote storage, including Windows Azure storage) based on requirements</p>	<p>Revised subtask: including Windows Azure storage and Azure Mobile Services</p>
<p>Retrieve data remotely Use XHR to retrieve web services; set appropriate HTTP verb for REST; handle progress of data requests; consume SOAP/WCF services; use WebSockets for bidirectional communication</p>	<p>Revised subtask: Use XHR or HttpClient to retrieve web services</p>
<p>Implement data binding Choose and implement data-bound controls, including WinJS.UI.ListView, to meet requirements; bind data to item templates such as WinJS.Binding.Template; bind data to controls by using data-win-control and data-win-bind; configure an iterator with data-win-options; enable filtering, sorting, and grouping data in the user interface</p>	<p>No change</p>
<p>Manage Windows authentication and authorization Retrieve a user's roles or claims; store and retrieve credentials by using the PasswordVault class; implement the CredentialPicker class</p>	<p>New subtasks: verify credential existence by using credential locker; store account credentials in app settings</p>
<p>Manage web authentication Use the Windows.Security.Authentication.Web namespace; set up OAuth2 for authentication; CredentialPicker; set up single sign-on (SSO); implement credential roaming; implement the WebAuthenticationBroker class</p>	<p>New subtask: support proxy authentication for enterprises</p>