70-484:
Essentials of developing Windows Store apps using C#

The following tables itemize changes to exam 70-484. These changes will be made on December 2, 2013, to include updates that relate to Windows 8.1 and Visual Studio 2013 tasks.

1. Design Windows Store apps (increase: 19% to 20%)

Tasks currently measured	Tasks to be added/changed in December 2013
Design the UI layout and structure Evaluate the conceptual design and 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	Added subtask: use the Hub App template
Design for separation of concerns (SOC) Plan the logical layers of your solution to meet app requirements; design loosely coupled layers; incorporate WinMD components	No change
Apply the MVVM pattern to your app design Design and implement the appropriate data model to support business entities; design your viewmodel to support your view based on your model; develop a view to meet data-binding requirements; create view models using INotifyPropertyChanged, ObservableCollection, and CollectionViewSource	No change
Design and implement Process Lifetime Management (PLM) Choose a state management strategy; handle the suspend event; prepare for app termination; handle the Resume event; handle the OnActivated event; check the ActivationKind and previous state	No change
Plan for an app deployment Plan a deployment based on Windows Store app certification requirements; prepare an app manifest	No change

(capabilities and declarations); sign an app; plan the requirements for an enterprise deployment	

2. Develop Windows Store apps (no change: 19%)

Tasks currently measured	Tasks to be added/changed in December 2013
Access and display contacts Call the ContactsPicker class; filter which contacts to display; display a set number of contacts; create and modify contact information; select specific contact data	No change
Design for charms and contracts Choose the appropriate charm based on app requirements; design your app in a charm- and contract-aware manner; configure app manifest for correct permissions	No change
Implement search Provide search suggestions using the SearchPane class; search for 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	Modified subtask: Provide search suggestions using the SearchPane class and SearchBox control New subtask: configure search contracts
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	New subtask: use web links and application links
Manage app settings and preferences Choose which app features are accessed in AppSettings; add entry points for AppSettings in the Settings window; create settings flyouts using the Popup control; add settings options to Popup; store and retrieve settings from the roaming app data store	Modified subtask: create settings flyouts Removed subtask: add settings options to Popup
None	New task (section): **Integrate media features** Support DDS images; implement video playback; implement XVP and DXVA; implement Text to Speech (TTS)

3. Create the user interface (no change: 21%)

Tasks currently measured	Tasks to be added/changed in December 2013
Create layout aware apps to handle view states Handle view state events from ViewStateManager; choose between style patterns for the different view states; set app orientation in the manifest	Revised task: **Create layout-aware apps to handling windowing modes** Revised, complete list of subtasks: Respond to changes in orientation; adapt to new windowing modes by using the ViewManagement namespace; manage settings for an apps view
Implement layout controls Implement the Grid control to structure your layout; set the number of rows/columns and size; enable zoom and scroll capabilities in layout controls; manage text flow and presentation	No change
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	Revised task: **Design and implement the app/nav bar** Revised, complete list of subtasks: Determine what to put on the app/nav bar based on app requirements; style and position app/nav bar items; design the placement of controls on the app/nav bar; handle app/nav bar events
Design and implement data presentation Choose and implement data controls to meet app requirements (e.g., ListView, GridView, and FlipView); create data templates to meet app requirements	Revised subtask: Choose and implement data controls and properties to meet app requirements (e.g., ListView, GridView, FlipView, DatePicker, TimePicker, Hyperlink, PlaceholderText, menu flyouts, and CommandBar)
Create and manage XAML styles and templates Implement and extend styles and templates; implement gradients; modify styles based on event and property triggers; create shared resources and themes	No change

4. Program user interaction (decrease: 21% to 20%)

Tasks currently measured	Tasks to be added/changed in December 2013
Create and manage tiles Create and update tiles and tile contents; create and update badges (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 using the ToastUpdateManager class; play sounds with toast notifications; respond to toast events; control toast duration	Added subtask: configure and use Azure Mobile Services for push notifications
Manage input devices Capture gesture library events; create custom gesture recognizers; listen to mouse events or touch gestures; manage Stylus input and inking	No change
Design and implement navigation in an app Handle navigation events, check navigation properties, and call navigation functions by using the Navigation framework; design navigation to meet app requirements; Semantic Zoom	No change

5. Manage security and data (no change: 20%)

Tasks currently measured	Tasks to be added/changed in December 2013
Choose an appropriate data access strategy Choose the appropriate data access strategy (file based; web service; remote storage, including Windows Azure storage) based on requirements	Modified subtask: Choose the appropriate data access strategy (file based, web service, remote storage, including Windows Azure storage and Azure Mobile Services) based on requirements
Retrieve data remotely Use HttpClient to retrieve web services; set the appropriate HTTP verb for REST; consume SOAP/WCF services; use WebSockets for bi-directional communication; handle the progress of data requests	No change
Implement data binding Choose and implement data-bound controls; bind collections to items controls; implement the IValueConverter interface; create and set dependency properties; validate user input; enable filtering, grouping, and sorting data in the user interface	No change
Manage Windows Authentication Retrieve a user's roles or claims; store and retrieve credentials by using the PasswordVault class; implement the CredentialPicker class	Revised task: **Manage Windows authentication and authorization** Additional subtasks: verify credential existence by using credential locker; store account credentials in app settings
Manage Web Authentication Use the Windows.Security.Authentication.Web namespace; set up OAuth2 for authentication; implement the CredentialPicker class; set up single sign-on (SSO); implement credential roaming; implement the WebAuthenticationBroker class	Additional subtask: support proxy authentication for enterprises