

# Microsoft

## Exam 70-488

### Developing Microsoft SharePoint Server 2013 Core Solutions

Version: Demo

[ Total Questions: 10 ]

---

**Microsoft 70-488 : Practice Test**

---

**Topic break down**

<b>Topic</b>	<b>No. of Questions</b>
<b>Topic 2: SharePoint</b>	<b>1</b>
<b>Topic 3: Coho Winery</b>	<b>1</b>
<b>Topic 5: Mix Questions</b>	<b>8</b>

## Topic 2, SharePoint

A client asks you to create a basic task tracking app for SharePoint. The app will be hosted remotely. The users will be able to set up task lists, mark tasks as completed, and manage their task lists.

### Business Requirements

The client specifies the following business requirements:

- ✍ Users can create task lists.
- ✍ Users can retrieve task lists.
- ✍ Project leads can remove task lists.
- ✍ Users can modify task lists.
- ✍ Only users with a paid license can use the app.
- ✍ Tasks can be tagged with the programming language being used. Task list items may be tagged only with the following terms: C++, C#, and JavaScript.
- ✍ Navigational links across the top should be driven by the term store and use URLs that will improve search engine ranking.
- ✍ Set a system message that all users can access without the need to query a data source.
- ✍ Remote users will authenticate to the app by using a username and password.
- ✍ During development, the app will be tested by deploying to a development SharePoint site at <http://sharepoint.contoso.com/tasks>.
- ✍ When completed, the app will be published to the Microsoft Store as a paid app.

### Existing Site Configuration

The table below shows the current site configuration for the client's SharePoint site.

Site/Site Collection	Description
<a href="http://contoso-public.sharepoint.com">http://contoso-public.sharepoint.com</a>	Public-facing site
<a href="https://contoso.sharepoint.com/tasks">https://contoso.sharepoint.com/tasks</a>	Development site for tasks app
<a href="https://contoso.sharepoint.com/search">https://contoso.sharepoint.com/search</a>	Search site
<a href="https://contoso-my.sharepoint.com">https://contoso-my.sharepoint.com</a>	Personal My Site

### Task List Design

The app should create a SharePoint list named Tasks with the following attributes:

Field Name	Data Type	Contents
Department	Text	
Description	Text	
Hours	Number	
Completed	Radio Button	"Yes", "No"

### User Groups

SharePoint groups will be used to manage access to features of the app.

Group	Description
Admin	Admin can perform all tasks including creating projects, assigning users to projects, and maintaining all task lists.
Project Leads	Project leaders are the only members who have the ability to delete tasks.
Users	Users have the ability to add and manage tasks in their own projects, but cannot delete tasks.

### Task Display

The app should retrieve the Tasks list and display the contents on a webpage 20 tasks at a time.

### Technical Requirements

You have the following requirements:

- ✍ Query the Microsoft apps licensing service to validate the license.
- ✍ Package and configure the app.
- ✍ Configure navigation to use taxonomy-based navigation automatically generated by SharePoint.
- ✍ Utilize CSOM to retrieve data from the SharePoint site.
- ✍ Implement the system message by using a property bag that uses the key named SystemMessage.
- ✍ Implement and configure the credential store.
- ✍ The app will be internally tested for at least one month to track and fix any run-time errors.

### Pending Issues

When you deploy the app from Visual Studio 2012, you receive the error: "Error occurred in deployment step 'Install app for SharePoint': Sideloaded apps are not enabled on this site."

### Application Structure

Relevant portions of the app files are shown below. (Line numbers in the code segments are included for reference only and include a two-character prefix that denotes the specific file to which they belong.)

### LicenseValidation.cs

---

**Microsoft 70-488 : Practice Test**

---

```
LV01 protected bool IsLicenseValid(Guid productId, ClientContext context)
LV02 {
LV03     string token = "";
LV04     ClientResult<AppLicenseCollection> licenseCollection =
LV05     Utility.GetAppLicenseInformation(context, productId);
LV06     context.Load(context.Web);
LV07     context.ExecuteQuery();
LV08     foreach (AppLicense license in licenseCollection.Value)
LV09     {
LV10         token = license.RawXMLLicenseToken;
LV11         break;
LV12     }
LV13     VerificationServiceClient service = null;
LV14     VerifyEntitlementTokenResponse result = null;
LV15     VerifyEntitlementTokenRequest request =
LV16     new VerifyEntitlementTokenRequest();
LV17     request.EntitlementToken = token;
LV18     service = new VerificationServiceClient();
LV19     result = service.VerifyEntitlementToken(request);
LV20     if (result != null && result.AssetId != null)
LV21         return false;
LV22 }
```

## SystemMessage.cs

```
SM01 public void SetSystemMessage(string message)
SM02 {
SM03     string url = "https://sharepoint.contoso.com/tasks";
SM04     string property = "SystemMessage";
SM05     var context = new ClientContext(url);
SM06     NetworkCredential credentials =
SM07     new NetworkCredential("", "", "");
SM08     context.Credentials = credentials;
SM09     Site site = context.Site;
SM10     context.Load(site);
SM11     Web web = site.RootWeb;
SM12     context.Load(web, w => w.AllProperties);
SM13     context.ExecuteQuery();
SM14     PropertyValues propertyValues = web.AllProperties;
SM15     context.Load(propertyValues);
SM16     web.AllProperties[property] = message;
SM17     web.Update();
SM18 }
```

## ImplementTaskList.cs

---

**Microsoft 70-488 : Practice Test**

---

```

IL01 protected void ImplementTaskList()
IL02 {
IL03     ClientContext context = new ClientContext("http://sharpoing.contoso.com/tasks");
IL04     List list = context.Web.Lists.GetByTitle("Announcements");
IL05     Field department =
IL06         list.Fields.AddFieldAsXml("<Field DisplayName='Department' Type='Text' />",
IL07         true,
IL08         AddFieldOptions.DefaultValue);
IL09     FieldText departmentField = context.CastTo<FieldText>(department);
IL10     departmentField.Update();
IL11     Field description =
IL12         list.Fields.AddFieldAsXml("<Field DisplayName='Department' Type='Text' />",
IL13         true,
IL14         AddFieldOptions.DefaultValue);
IL15     FieldText descriptionField = context.CastTo<FieldText>(description);
IL16     descriptionField.Update();
IL17     Field hours =
IL18         list.Fields.AddFieldAsXml("<Field DisplayName='Hours' Type='Number' />",
IL19         true, AddFieldOptions.DefaultValue);
IL20     FieldNumber hoursField = context.CastTo<FieldNumber>(hours);
IL21     hoursField.Update();
IL22
IL23     context.ExecuteQuery();
IL24 }

```

**RetrieveTaskList.cs**

```

RT01 protected void RetrieveTaskList()
RT02 {
RT03     ClientContext context = new
RT04         ClientContext("https://sharepoint.contoso.com/tasks");
RT05     List taskList = context.Web.Lists.GetByTitle("Tasks");
RT06     CamlQuery query = CamlQuery.CreateAllItemsQuery(20);
RT07     Microsoft.SharePoint.Client.ListItemCollection
RT08         tasks = taskList.GetItems(query);
RT09     context.Load(tasks);
RT10     context.ExecuteQuery();
RT11     foreach (Microsoft.SharePoint.Client.ListItem task in tasks)
RT12     {
RT13         ...
RT14     }
RT15 }

```

**CanDeleteTask.cs**

```

CD01 protected bool CanDeleteTask(string targetEmail)
CD02 {
CD03     ClientContext context = new ClientContext("https://sharepoint.contoso.com/
CD04     tasks");
CD05     context.ExecuteQuery();
CD06
CD07     foreach (User user in members.Users)
CD08     {
CD09         if (user.Email == targetEmail)
CD10         {
CD11             return true;
CD12         }
CD13     }
CD14     return false;
CD15 }

```

**TermSet Navigation.cs**



## Microsoft 70-488 : Practice Test

```
TN01 public void CreateNavigationTermSet()
TN02 {
TN03     ClientContext clientContext = new ClientContext("https://sharepoint.contoso.com/
tasks");
TN04     TaxonomySession session = TaxonomySession.GetTaxonomySession(clientContext);
TN05     taxonomySession.UpdateCache();
TN06     clientContext.Load(session, ts => ts.TermStores);
TN07     clientContext.ExecuteQuery();
TN08     TermStore store = taxonomySession.TermStores[0];
TN09     clientContext.Load(store,
TN10         s => s.Name,
TN11         s => s.WorkingLanguage);
TN12     TermGroup siteCollectionGroup = termStore.GetSiteCollectionGroup
(clientContext.Site, createIfMissing: true);
TN13     TermSet set = siteCollectionGroup.CreateTermSet("Task Terms");
TN14     store.CommitAll();
TN15     clientContext.ExecuteQuery();
TN16
TN17     termSet.IsNavigationTermSet = true;
TN18     termSet.TargetUrlForChildTerms.Value = "~site/Pages/Tasks/Tasks.aspx";
TN19     termStore.CommitAll();
TN20     clientContext.ExecuteQuery();
TN21     NavigationTerm term = termSet.CreateTerm("Programming
Languages", NavigationLinkType.SimpleLink,
TN22         Guid.NewGuid());
TN23     term.SimpleLinkUrl = "https://sharepoint.contoso.com/tasks";
TN24
TN25     term.GetTaxonomyTerm().TermStore.CommitAll();
TN26     clientContext.ExecuteQuery();
TN27 }
```

### Question No : 1 - (Topic 2)

You need to implement term set navigation.

Which code segment should you insert at line TN16?

## Microsoft 70-488 : Practice Test

---

- A. `NavigationTermSet navigationTerms =  
NavigationTermSet.GetAsResolvedByWeb(clientContext,  
termSet, clientContext.Web,  
"GlobalNavigationTaxonomyProvider");`
- B. `NavigationTermSet navigationTerms =  
NavigationTermSet.GetAsResolvedByWeb(clientContext,  
termSet, clientContext.Web,  
"GlobalNavigationSwitchableProvider");`
- C. `NavigationTermSet navigationTerms =  
NavigationTermSet.GetAsResolvedByWeb(clientContext,  
termSet, clientContext.Web,  
"CurrentNavigationTaxonomyProvider");`
- D. `NavigationTermSet navigationTerms =  
NavigationTermSet.GetAsResolvedByWeb(clientContext,  
termSet, clientContext.Web,  
"CurrentNavigationSwitchableProvider");`

- A. Option A  
B. Option B  
C. Option C  
D. Option D

### Answer: A

**Explanation:** \* A well-designed navigation tells your site's users a lot about the business, products, and services that the website offers. By updating the taxonomy behind the navigation, businesses can drive and keep up with change without having to recreate their site navigation in the process.

\* GlobalNavigationToxonomyProvider

StandardNavigationProviderNames.GlobalNavigationTaxonomyProvider field

The TaxonomySiteMapProvider instance used for the global navigation scheme.

The menu for the "global navigation" scheme typically appears at the top of the master page.

Scenario: Navigational links across the top should be driven by the term store and use URLs that will improve search engine ranking.

Incorrect:



Not C: StandardNavigationProviderNames.CurrentNavigationTaxonomyProvider field  
The menu for the “current navigation” scheme typically appears on the left side of the master page.

Reference: Managed navigation in SharePoint 2013;  
StandardNavigationProviderNames.GlobalNavigationTaxonomyProvider field.

### Topic 3, Coho Winery

#### Background

Coho Winery implements a SharePoint farm. They have an intranet site for company employees and an extranet site for their customers. Coho Winery sells products to wholesale and retail customers. Customers must be able to order specific brands and pay directly.

#### Business Requirements

##### Site Collection

The company must organize its main enterprise site to have multiple site collections for different wine brands from various regions. Site documents must have a consistent look and feel.

##### New Features

The first version (1.0.0.0) of the feature must support the addition of newer functionality when the feature is upgraded to version 1.5.0.0.

#### Technical Requirements

##### General

- ✍ A Document content type named Finance must be created at the site level. It must contain two site columns named Customer Name and Transaction Date.
- ✍ The Order Details and Payment Confirmation content types must be derived from the Finance content type.
- ✍ Workflows associated with document content types must be initiated when documents are generated.
- ✍ When Payment Confirmation documents are generated, a workflow must send a notification to the customer by email.

##### User Authorization

- ✍ A custom role named Wholesale must be created. Wholesale customers must be assigned to this role in order to view the promotions page.
- ✍ A custom permission level must be created. This permission level must add the ViewPages permission to the existing base permissions. This permission level must be added to Wholesale user role.

- ✍ Anonymous users must be able to browse the site with limited access to resources.
- ✍ The web application has been created with anonymous access enabled.
- ✍ Users are prompted for credentials when they browse to the site.

### **Application Lifecycle Management**

- ✍ Versioning of the solution artifacts must adhere to industry best practices.
- ✍ In a future upgrade process, a new site column named Division must be added to the Finance content type programmatically.
- ✍ Only one version of an assembly must be available at runtime.
- ✍ Code based hot fixes must be deployed directly to the Global Assembly Cache on all SharePoint servers.
- ✍ During maintenance periods, content must be available but must not be modified.

### **Backup and Recovery**

- ✍ In the event of hardware failure, the SharePoint environment must be restored to the most recent date.
- ✍ The backup process must not degrade system performance.

### **Performance Optimization and Memory Usage**

SPSite objects must be removed from memory immediately after the objects go out of scope.

### **Automation**

Windows PowerShell must be used to automate all operations wherever possible.

### **Application Structure**

Relevant portions of the application files are shown below. (Line numbers in the code segments are included for reference only and include a two-character prefix that denotes the specific file to which they belong.)

#### **ContentType.cs**

```
CT01 public override void
    FeatureActivated(SPFeatureReceiverProperties spFtRcvProperties)
CT02 {
CT03     SPWeb web =
        ((SPSite)spFtRcvProperties.Feature.Parent).RootWeb;
CT04     string financeColumn = "Finance Columns";
CT05     string customerFieldName =
        web.Fields.Add("Customer Name", SPFieldType.Text, true);
CT06
CT07     string transDateFieldName = web.Fields.Add("Transaction
Date", SPFieldType.DateTime, true);
CT08     ...
CT09     string contentTypeGroup = "Finance Content Types";
CT10     SPContentType docContentType =
        web.AvailableContentTypes[SPBuiltInContentTypeId.Document];
CT11
CT12     docContentType1 = web.ContentTypes.Add(docContentType1);
CT13     docContentType1.Group = contentTypeGroup;
CT14     SPFieldLink transDateFieldRef =
        new SPFieldLink(transDateField);
CT15     transDateFieldRef.Required = true;
CT16     docContentType1.FieldLinks.Add(transDateFieldRef);
CT17     docContentType1.Update();
CT18     docContentType2 = web.ContentTypes.Add(docContentType2);
CT19     docContentType2.Group = contentTypeGroup;
CT20     SPFieldLink titleODFieldRef =
        docContentType2.FieldLinks[SPBuiltInFieldId.Title];
CT21     ...
CT22     titleODFieldRef.Required = true;
CT23     ...
CT24     docContentType2.Update();
CT25     docContentType3 = web.ContentTypes.Add(docContentType3);
CT26     docContentType3.Group = contentTypeGroup;
CT27     SPFieldLink titlePmtFieldRef =
        docContentType3.FieldLinks[SPBuiltInFieldId.Title];
CT28     ...
CT29     titlePmtFieldRef.Required = true;
CT30     ...
CT31     docContentType3.Update();
CT32 }
CT33
CT34 {
CT35     SPWeb parentWeb = (SPWeb)properties.Feature.Parent;
CT36     switch (upgradeActionName)
CT37     {
CT38         case "                ":
CT39             ...
CT40             break;
CT41
CT42         default:
CT43             ...
CT44             break;
CT45     }
CT46 }
```

```
UA01 private SPBasePermissions _permissions =
    SPBasePermissions.EmptyMask;
UA02 public override void
    FeatureActivated(SPFeatureReceiverProperties properties)
UA03 {
UA04     try
UA05     {
UA06         SPSite site =
    new SPSite("http://localhost:80/sites/brands/Wholesale");
UA07
UA08         using (SPWeb web = site.RootWeb)
UA09         {
UA10             _permissions = this.GetExistingPermissions();
UA11
UA12             foreach (SPWeb spWebItem in web.Webs)
UA13             {
UA14                 ...
UA15             }
UA16         }
UA17     }
UA19     {
UA20
UA21     }
UA22
UA23 }
UA24 private SPBasePermissions GetExistingPermissions()
UA25 {
UA26     SPBasePermissions permissions =
    SPBasePermissions.EmptyMask;
UA27     ...
UA28     return permissions;
UA29 }
```

**ContentType.Finance.Template.xml**

---

**Microsoft 70-488 : Practice Test**


---

```

CX01 <Feature Id="08380D6A-858C-4618-9525-50E9D6DEAD53"
CX02   Title="ContentType.Finance"
CX03   Description="Content type for Finance"
CX04   ReceiverClass="MyFeatureReceiver.MyReceiver"
CX05   xmlns="http://schemas.microsoft.com/sharepoint/"
CX06   ...
CX07   <ElementManifests>
CX08     <ElementManifest Location="ElementsV1.xml" />
CX09     <ElementManifest Location="ElementsV2.xml" />
CX10   </ElementManifests>
CX11
CX12   <UpgradeActions>
CX13
CX14     <ApplyElementManifests>
CX15       <ElementManifest Location="ElementsV2.xml" />
CX16     </ApplyElementManifests>
CX17     <AddContentTypeField
CX18       ContentTypeId="0x010123F15ADB2FA333AA49848E7E01BC79C9753222"
CX19       FieldId="{4FB9816F-799D-41F3-B801-B8F5FEB3A5CD}"/>
CX20       <CustomUpgradeAction Name="          "
CX21         <Parameters>
CX22           <Parameter Name="Division">Finance</Parameter>
CX23         </Parameters>
CX24       </CustomUpgradeAction>
CX25
CX26   </UpgradeActions>
CX27
CX28 </Feature>

```

### ExportWholesaleSites.ps1

```

EW01 $site = Get-SPSite "http://localhost:80/sites/brands/Wholesale";
EW02 foreach ($web in $site.AllWebs)
EW03 {
EW04
EW05     $web | Export-SPWeb -Path ($web.Title + ".cmp") -Force;
EW06     $date = Get-Date;
EW07     if ($web.AllProperties["LastWebExport"] -eq $null)
EW08     {
EW09         $web.AddProperty("LastWebExport", $date);
EW10         $web.Update();
EW11     }
EW12     else
EW13     {
EW14         $web.SetProperty("LastWebExport", $date);
EW15         $web.Update();
EW16     }
EW17
EW18 }

```

**Question No : 2 - (Topic 3)**

You must upgrade the existing version of the application to a newer version.



Which two actions should you perform? (Each correct answer presents part of the solution. Choose all that apply.)

A. Add the following code segment at line CX11:

```
<VersionRangeBeginVersion="1.0" EndVersion="2.e">
```

B. Add the following code segment at CX13:

```
<VersionRange BeginVersion-"1.0.0.0"EndVersion-"2.0.0.0">
```

C. Use Microsoft Visual Studio to change the feature version to 1.5.0.0.

D. Use Microsoft Visual Studio to change the feature version to 1.5.

E. Add the following code segment at line CX11:

```
<VersionRange BeginVersion-"1.0.0.0"EndVersion-"2.0.0.0">
```

**Answer: B,C**

**Explanation:** B: VersionRange Element specifies a range of previous versions of the Feature to which the upgrade actions should apply.

/ BeginVersion: Specifies the earliest version number to which the child upgrade elements will apply, in the format n.n.n.n, where each n can be up to four digits.

/ EndVersion: Specifies the earliest version number to which the upgrade will not apply. The format is n.n.n.n, where each n can be up to four digits.

C: Feature versions in Microsoft SharePoint Foundation are used to trigger Feature upgrade. If you update your Feature version but do not specify any Feature upgrade logic in the Feature.xml file, when the Feature upgrade runs, it simply updates the version number of your Feature. However, if you specify Feature upgrade logic, the upgrade code is executed and the version number is automatically updated as well.

Scenario: New Features

The first version (1.0.0.0) of the feature must support the addition of newer functionality when the feature is upgraded to version 1.5.0.0.

Reference: VersionRange Element (Feature); Best Practices for Using Feature Versions

## Topic 5, Mix Questions

### Question No : 3 DRAG DROP - (Topic 5)

You work on a SharePoint Enterprise Content Management (ECM) project.



## Microsoft 70-488 : Practice Test

You need to ensure that all Microsoft Word documents are converted to Portable Document Format (PDF) once they are published and approved.

You have the following code:

```
public override void Target 1(SPItemEventProperties properties)
{
    base.Target 2(properties);
    SPListItem spListItem = properties.ListItem;
    SPFile spFile = spListItem.File;
    String spFileExtension = Path.GetExtension(spFile.Name);
    if ((spFileExtension.ToUpper() == ".DOCX") &&
        (spFile.Level == Target 3.Target 4))
    {
        ConversionJob conversionJob =
            new ConversionJob("Word Automation Services Application");
        conversionJob.Name = "PDF Conversion";
        conversionJob.UserToken = properties.Web.CurrentUser.UserToken;
        conversionJob.Settings.OutputSaveBehavior =
            SaveBehavior.AlwaysOverwrite;
        conversionJob.Settings.OutputFormat = SaveFormat.PDF;
        conversionJob.Settings.UpdateFields = true;
        string siteUrl = properties.WebUrl + "/";
        string outputUrl = siteUrl+spFile.Url.Replace(".docx", ".pdf");
        conversionJob.AddFile(siteUrl + spFile.Url, outputUrl);
        conversionJob.Start();
    }
}
```

Which code segments should you include in Target 1, Target 2, Target 3 and Target 4 to complete the code? (To answer, drag the appropriate code segments to the correct targets in the answer area. Each code segment may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content)

Code Segments	Answer Area
Approved	Target 1: Code Segment
ItemAdded	Target 2: Code Segment
ItemUpdated	Target 3: Code Segment
Published	Target 4: Code Segment
Run();	
SPFileInfo	
SPFileLevel	
Start();	

Answer:

Code Segments	Answer Area
Approved	Target 1: ItemUpdated
ItemAdded	Target 2: ItemUpdated
ItemUpdated	Target 3: SPFileLevel
Published	Target 4: Published
Run();	
SPFileInfo	
SPFileLevel	
Start();	

Explanation:

Target 1:

ItemUpdated

Target 2:

ItemUpdated

Target 3:

SPFileLevel

Target 4:

Published

\* ItemUpdated

Runs after an item has been updated in the list

\* SPFileLevel.Published

SPFileLevel specifies the level of publication for a document.

Published: The document is published.

**Question No : 4 DRAG DROP - (Topic 5)**

You develop a SharePoint app to access data contained in a hosted SharePoint site.

You need to authorize the app to access the list items in a custom list on the host website.

You have the following code:

```
Target 1.TrustAllCertificates();
string Target 2 =
    Target 3.GetContextTokenFromRequest(Request);
if (contextTokenStr != null)
{
    contextToken = Target 4.ReadAndValidateContextToken
        (Target 5 , Request.Url.Authority);
    sharepointUrl = new Uri(Request.QueryString["SPHostUrl"]);
    accessToken = TokenHelper.Target 6
        (contextToken,sharepointUrl.Authority).AccessToken;
    Button1.CommandArgument = accessToken;
}
```

Which code segments should you include in Target 1, Target 2, Target 3, Target 4, Target 5 and Target 6 to complete the code? (To answer, drag the appropriate code segments to the correct location or locations in the answer area. Each code segment may be used once, more than once, or not at all. You may need to drag the split bar between panes or scroll to view content)

Code Segments	Answer Area
<input type="text" value="ServiceHost"/>	Target 1: <input type="text" value="Code Segment"/>
<input type="text" value="SecurityTokenContext +"/>	Target 2: <input type="text" value="Code Segment"/>
<input type="text" value="TokenHelper"/>	Target 3: <input type="text" value="Code Segment"/>
<input type="text" value="GetAccessToken"/>	Target 4: <input type="text" value="Code Segment"/>
<input type="text" value="contextTokenStr"/>	Target 5: <input type="text" value="Code Segment"/>
	Target 6: <input type="text" value="Code Segment"/>

Answer:

The screenshot shows a code editor interface with two main sections: 'Code Segments' on the left and 'Answer Area' on the right. In the 'Code Segments' section, five code snippets are listed, each enclosed in a dashed green border: 'ServiceHost', 'SecurityTokenContext', 'TokenHelper', 'GetAccessToken', and 'contextTokenStr'. In the 'Answer Area' section, six targets are listed, each with a corresponding code snippet in a dashed red border: Target 1: TokenHelper, Target 2: contextTokenStr, Target 3: TokenHelper, Target 4: TokenHelper, Target 5: contextTokenStr, and Target 6: GetAccessToken.

**Explanation:**

The diagram shows six targets, each with a corresponding code snippet in a dashed red border: Target 1: TokenHelper, Target 2: contextTokenStr, Target 3: TokenHelper, Target 4: TokenHelper, Target 5: contextTokenStr, and Target 6: GetAccessToken.

\* (Target1) TokenHelper.TrustAllCertificates

If you are using Visual Studio 2012, it adds 'TrustAllCertificates' in TokenHelper.cs by default.

\* (target3) Tokenhelper.GetContextTokenFromRequest

\* (Target2, Target4, target5)Tokenhelper.ReadAndValidateContextToken

Example:

```
string contextTokenString = TokenHelper.GetContextTokenFromRequest(Page.Request);  
SharePointContextToken contextToken =  
TokenHelper.ReadAndValidateContextToken(contextTokenString, Request.Url.Authority);
```

\*(Target6) TokenHelper.GetAccessToken

Example:

```
//Get user+app access token.
```

```
string accessToken =
```

```
TokenHelper.GetAccessToken(contextToken, sharepointUrl.Authority).AccessToken;
```

### Question No : 5 - (Topic 5)

You are developing a SharePoint solution for a company. You create one site collection for each department. Sites may use data from internal or external web service endpoints. You are developing an application that allows site owners to add or change web service endpoints.

The application must run in the context of the site owner.

You need to complete the application.

What should you do?

- A.** Use the SPWebConfigModification class to add a list of web service endpoint URLs to the web.config file.
- B.** Add a property named WebSvcEndpoint to the SPSite.Properties property bag. Store a web service endpoint URL as a value.
- C.** Add a property named WebSvcEndpoint to the SPWeb.AllProperties property bag. Store a web service endpoint URL as a value.
- D.** Add a list of web service endpoint URLs to the SPFarm.Properties hashtable.

**Answer: A**

**Explanation:** The SPWebConfigModification class holds modifications that are made to the web.config.

Each endpoint supported by your WCF service application must be defined in the web.config settings for the application.

Reference: Using Service Endpoints

[https://msdn.microsoft.com/en-us/library/office/ee535060\(v=office.14\).aspx](https://msdn.microsoft.com/en-us/library/office/ee535060(v=office.14).aspx)



**Question No : 6 - (Topic 5)**

Adventure Works uses a SharePoint publishing site to host their public-facing website at <http://www.adventureworks.com>. The website gives external users the ability to register and sign in to the site to buy Adventure Works products.

You notice that publicly registered users see the SharePoint ribbon.

You need to ensure that the SharePoint ribbon is available only to content authors and administrators.

What should you do?

- A. In the SharePoint project, create a User Control and add the following code segment to hide the ribbon:

```
protected void Page_Load(object sender, EventArgs e)
{
    if (UserHasPermissions(SPContext.Current.Web.CurrentUser))
    {
        SPRibbon.GetCurrent(this.Page).CommandUIVisible = true;
    }
    else
    {
        SPRibbon.GetCurrent(this.Page).CommandUIVisible = false;
    }
}
private bool UserHasPermissions(SPUser sPUser)
```

Then place the control in the master page.
- B. Open the current site master page in SharePoint Designer, place the SharePoint:SPSecurityTrimmedControl control around the div tag with the s4-ribbonrow ID. Set the PermissionsString to **ManageSubWeb**.
- C. Open the current site master page in SharePoint Designer, find a div tag with the s4-ribbonrow ID, and then set the style attribute to **display:none**.
- D. In the SharePoint project, create a User Control and add the following code segment to hide the ribbon:

```
protected void Page_Load(object sender, EventArgs e)
{
    if (UserHasPermissions(SPContext.Current.Web.CurrentUser))
    {
        SPContext.Current.Site.CommandUIVisible = true;
    }
    else
    {
        SPContext.Current.Site.CommandUIVisible = false;
    }
}
private bool UserHasPermissions(SPUser sPUser)
```

Then place the control in the master page.

- A. Option A
- B. Option B
- C. Option C
- D. Option D

**Answer: A**

**Explanation:** The property CommandUIVisible shows or hides the ribbon section at the top of the application page.

Incorrect:

Not B: Permission should be set to AddAndCustomizePages not ManageSubWeb.

Reference: SharePoint, Hide the Ribbon from Anonymous Users

<http://www.topsharepoint.com/hide-the-ribbon-from-anonymous-users>

**Question No : 7 - (Topic 5)**

You plan to create a workflow design by using Microsoft Visio 2013 and then import the design into SharePoint Designer 2013.

You need to ensure that you have branching capability.

Which shape should you use?

- A. Assign a Task
- B. Stage
- C. Loop
- D. Step

**Answer: B**

**Explanation:** A stage can contain any number of shapes and may include branching.

**Question No : 8 - (Topic 5)**

You must select an app model to deploy apps to an Office 365 small business tenant. The business logic of the apps is located in a remote web application.

What should you do?

- A. Create a SharePoint-hosted app. Implement the business logic by using JavaScript.
- B. Develop provider-hosted apps and integrate data sources with the business logic by using OData.
- C. Deploy the business logic to the host web by using a Full Trust SharePoint solution.
- D. Develop an OData service provider infrastructure and publish the provider to the \_layouts directory.

**Answer: B**

**Explanation:** Provider Hosted Application

Provider hosted apps are also a part of cloud hosting model, where app can be hosted on any remote web platform such as IIS server. And these apps are made available on deploying it on cloud i.e. using SharePoint online, Office365 site, or windows azure in case of Auto hosting.

It interacts with SharePoint 2013 by using one of the SharePoint client object models or its REST/OData-based web service. Depending on the scenario it gains authorization to SharePoint data using either the SharePoint 2013 implementation of the OAuth standard or a JavaScript cross-domain library.

Note: Apps in SharePoint

SharePoint 2013 app model has three types of hosting model.

- a. SharePoint Hosted Application.
- b. Auto Hosted Application.
- c. Provider Hosted Application.

Incorrect:

Not C: When you use the full-trust execution model, you deploy your assemblies to the global assembly cache on each Web front-end server and application server in the server farm. The SharePoint Web application process loads the assembly from the global assembly cache and your code runs with full trust—in other words, it runs without any code access security restrictions.

Reference: Step by step create provider hosted application using SharePoint 2013

**Question No : 9 - (Topic 5)**

Adventure Works uses a SharePoint publishing site for an external website at [www.adventureworks.com](http://www.adventureworks.com). The user interface is implemented by using HTML5, CSS 3, and JavaScript.

You need to ensure that the site designers have an efficient way to create and modify the JavaScript files.

What should you do?

- A.** Open the site in SharePoint Designer. From All Files, double-click the \_cts folder and then select the MasterPages folder. Create a folder. Modify the JavaScript files by using Microsoft Notepad.
- B.** Copy all the JavaScript files to a document library and modify them directly by using Microsoft Notepad.
- C.** Open the site in SharePoint Designer. From Master Pages select the JavaScript files to modify them.
- D.** Set up a mapped network drive in Design Manager. Open the mapped drive and create a folder. Modify the JavaScript files by using Microsoft Notepad.

**Answer: C**

**Explanation:** Master Page - Use SharePoint Designer to add code to a master page when you want to code to be available on every page in a site. You can add the JavaScript inside of <SCRIPT> tags or link to a file that contains the JavaScript. Typically add your CSS just before the </HEAD> section and your JavaScript just before the </BODY> tag.

Reference: Adding JavaScript and CSS to SharePoint

**Question No : 10 DRAG DROP - (Topic 5)**

You plan to create a search-driven navigation that supports the Cross-site Publishing Feature.

Which four actions should you perform in sequence? (To answer, move the appropriate actions from the list of actions to the answer area and arrange them in the correct order.)

Microsoft 70-488 : Practice Test

Actions	Answer Area
Insert a Search Box Web Part on the page.	
Insert a Content Search Web Part on the page.	
Insert a Search Result Web Part on the page.	
Select the <b>Change query</b> button.	
Build and save the query.	
Select the <b>Edit Web Part</b> link.	

Answer:

Actions	Answer Area
Insert a Search Box Web Part on the page.	Insert a Content Search Web Part on the page.
Insert a Content Search Web Part on the page.	Select the <b>Edit Web Part</b> link.
Insert a Search Result Web Part on the page.	Build and save the query.
Select the <b>Change query</b> button.	Insert a Search Result Web Part on the page.
Build and save the query.	
Select the <b>Edit Web Part</b> link.	

Explanation:

Box 1: Insert a Content Search Web part on the page.

Box 2: Select the Edit Web Part link.

Box 3: Build and save the query.

Box 4: Insert a Search Results Web Part on the page.

Note:

Box 1: Add a Content Search Web Part to a page

Box 2-3: Configure the query for a Content Search Web Part

In the Web Part, click the Content Search Web Part Menu arrow, and then click Edit Web Part.

In the Web Part tool pane, in the Properties section, in the Search Criteria section, click Change query.

Edit and change the query.

Box 4: (Search Result Web Part)

Here are some tools that make Cross-Site Publishing happen.

/ Content Search Web Part:

It allows to build your own query using a wizard, this means fetching the content you want to see, and then allows you to display them using predefined display templates. It uses Search to query, which therefore allows us to show content from another Site.

/ Search Results Web Part:

It can either show the results of a typed query in the Search Box, which is the traditional Search Results we know. But, and this is key, the Search Results also allows you to build a Query using the same Query Builder in the Content Search. In fact, I have found the Search Results to offer 95% of the same functionality as the Content Search. Therefore, it can also be used in Cross-Site Publishing Scenarios.



**Thank You For Trying 70-488 PDF Demo**

**To try our 70-488 Premium Files visit link below:**

**<https://examsland.com/latest-exam-questions/70-488/>**

**Start Your 70-488 Preparation**

**Use Coupon **EL25** for extra 25% discount on the purchase of Practice Test Software.**