

# Microsoft

## Exam 70-541

### TS: Microsoft Windows SharePoint Services 3.0 – Application Development

Version: 43.0

[ Total Questions: 109 ]

**Question No : 1 - (Topic 0)**

You create a Microsoft Windows SharePoint Services application. A SharePoint site has a list

named MyList. You write the following code segment. (Line numbers are included for reference

only.)

```
01 SPWeb site = SPControl.GetContextWeb(context);
```

```
02 SPList list = site.Lists["MyList"];
```

```
03 ...
```

You need to bind the list to a DataGrid control. You also need to ensure that all the fields in the

list are available in the DataTable object. Which code segment should you insert at line 03?

**A.** `SPListItemCollection listItems = list.Items;`

`DataTable table = listItems.GetDataTable();`

**B.** `SPListItemCollection listItems = list.Items;`

```
foreach (SPListItem item in listItems) {  
    item.Properties["Visible"]=true;  
}
```

`DataTable table = listItems.GetDataTable();`

**C.** `SPListItemCollection listItems = list.Items;`

`DataTable table=new DataTable();`

```
foreach (SPListItem item in listItems) {  
    DataColumn col= new DataColumn(item.Name);  
    col.ExtendedProperties["Visible"]=true;  
    table.Columns.Add(col);  
}
```

**D.** `Guid dataViewGuid = list.Views["DataView"].ID;`

`SPView srcView = list.GetView(dataViewGuid);`

`SPViewFieldCollection viewFields = srcView.ViewFields;`

`SPListItemCollection listItems = list.GetItems(srcView);`

`DataTable table = listItems.GetDataTable();`

**Answer: A**

**Question No : 2 - (Topic 0)**

You create a Microsoft Windows SharePoint Services site.

You need to update the first item in a SharePoint list named MyList without triggering any alerts.

Which code segment should you use?

- A.** SPSite site = SPControl.GetContextSite(context);  
SPList destList = site.AllWebs[0].Lists[0];  
SPListItemCollection listItems = destList.Items;  
SPListItem item = listItems[0];  
item["Type"] = "Important";  
item["Assigned"] = "No";  
item.UpdateOverwriteVersion();
- B.** SPSite site = SPControl.GetContextSite(context);  
SPList destList = site.AllWebs["MySite"].Lists["MyList"];  
SPListItemCollection listItems = destList.Items;  
SPListItem item = listItems[0];  
item["Type"] = "Important";  
item["Assigned"] = "No";  
item.Recycle();
- C.** SPSite site = SPControl.GetContextSite(context);  
SPList destList = site.AllWebs["MySite"].Lists["MyList"];  
SPListItemCollection listItems = destList.Items;  
SPListItem item = listItems[0];  
item["Type"] = "Important";  
item["Assigned"] = "No";  
item.SystemUpdate();
- D.** SPWeb mySite = SPControl.GetContextWeb(context);  
SPListItemCollection listItems = mySite.Lists["MyList"].Items;  
SPListItem item = listItems.Add();  
item["Type"] = "Important" ;  
item["Assigned"] = "No";  
item.Update();

**Answer: C**

### Question No : 3 - (Topic 0)

You are writing a console application that displays information about attachments for list items.

You need to display the name and length of each attachment for an item.

Which code segment should you use?

**A.** `public static void DisplayAttachments ( SPListItem item)`

```
{  
foreach (string name in item.Attachments )  
{  
string fullName = item.Attachments.UrlPrefix + name;  
FileInfo fi = new FileInfo ( fullName );  
Console.WriteLine ("{0} has a length of {1}",  
fullName , fi.Length );  
}  
}
```

**B.** `public static void DisplayAttachments ( SPListItem item)`

```
{  
foreach (string name in item.Attachments )  
{  
SPFile spFile = item.Web.GetFile (name);  
Console.WriteLine ("{0} has a length of {1}", name, spFile.Length );  
}  
}
```

**C.** `public static void DisplayAttachments ( SPListItem item)`

```
{  
foreach (string name in item.Attachments )  
{  
string fullName = item.Attachments.UrlPrefix + name;  
SPFile spFile = item.Web.GetFile ( fullName );  
Console.WriteLine ("{0} has a length of {1}",  
fullName , spFile.Length );  
}  
}
```

**D.** `.public static void DisplayAttachments ( SPListItem item)`

```
{  
foreach (string name in item.Attachments )  
{  
FileInfo fi = new FileInfo (name);  
Console.WriteLine ("{0} has a length of {1}", name, fi.Length );  
}  
}
```

**Answer: C**

**Question No : 4 - (Topic 0)**

You create a Microsoft Windows SharePoint Services list named Books. The list has a column

named Stock that stores the total number of available books.

You write the following code segment. (Line numbers are included for reference only.)

```
01 protected override void CreateChildControls()
02 {
03 SPWeb myWeb = SPControl.GetContextWeb(Context);
04 SPList myList = myWeb .Lists["Books"];
05 ...
06 DataGrid myGrid = new DataGrid();
07 myGrid.DataSource = results;
08 myGrid.DataBind();
09 this.Controls.Add(myGrid);
10 }
```

You need to render a list of books for which the value in the Stock column is less than 100.

Which code segment should you insert at line 05?

- A.** SPQuery query = new SPQuery();  
query.Query = "<Where><Leq><FieldRef Name='Stock'/><Value Type='Number'>100</Value></Leq></Where>";  
SPListItemCollection results = list.GetItems(query);
- B.** SPQuery query = new SPQuery();  
query.Query = "<Where><Leq><FieldRef Name='Stock'/><Value Type='Number'>100</Value></Leq/Where>";  
SPListItemCollection results = list.GetChanges(query);
- C.** SPQuery query = new SPQuery();  
query.Query = "<Where><Lt><FieldRef Name='Stock'/><Value Type='Number'>100</Value></Lt></Where>";  
SPListItemCollection results = list.GetItems(query);
- D.** SPQuery query = new SPQuery();  
query.Query = "<Where><Lt><FieldRef Name='Stock'/><Value Type='Number'>100</Value></Lt></Where>";  
SPListItemCollection results = list.GetChanges(query);

**Answer: C**

**Question No : 5 - (Topic 0)**

You create a Microsoft Windows SharePoint Services site that contains task lists.

You are creating a custom Search Web Part that displays the number of unassigned tasks on the site.

You need to retrieve a list of task items from within the Web Part for which the assignment field is empty.

What should you do?

- A.** Create an instance of the FullTextSqlQuery class and pass a SQL string to the QueryText property.
- B.** Create an instance of the KeywordQuery class and set the QueryText property to the value AssignedTo.
- C.** Create an instance of the SPQuery class and pass a Collaborative Application Markup Language (CAML) string to the query property. Iterate through the task lists on the site and pass the query property to the SPList.GetItems method.

**Page 5 of 55**

- D.** Create an instance of the SPQuery class and pass a Collaborative Application Markup Language (CAML) string to the query property. Iterate through the task lists on the site and pass the SPQuery object to the SPList.GetItems method.

**Answer: D**

**Question No : 6 - (Topic 0)**

You create a Microsoft Windows SharePoint Services site on a remote server.

The site contains a records repository.

You create a Microsoft Windows Forms application that stores official documents in the records

repository by using Records Repository Web Service.

You need to submit a document to the records repository.

Which two actions should you perform? (Each correct answer presents part of the solution.

Choose two.)

- A. Serialize the document to an XML string.
- B. Impersonate a user with EditListItems permissions.
- C. Create a byte array that contains the document content.
- D. Declare an array of OfficialFileProperty objects to specify the metadata that is being submitted along with the document.

**Answer: C,D**

### Question No : 7 - (Topic 0)

You create a Microsoft Windows SharePoint Services site. You add a recurring event to an Events list by using the RecurrenceData property of the SPLListItem class.

You need to ensure that the event occurs once every Monday for five consecutive weeks.

Which XML fragment should you use?

- A. `< recurrence > < rule > < firstDayOfWeek > su < /firstDayOfWeek > < weekly mo='TRUE' weekFrequency='1' / > < repeatInstances > 5 < /repeatInstances > < /rule > < /recurrence >`
- B. `< recurrence > < rule > < firstDayOfWeek > su < /firstDayOfWeek > < weekly mo='TRUE' weekFrequency='5' / > < /rule > < /recurrence >`
- C. `< recurrence > < rule > < firstDayOfWeek > su < /firstDayOfWeek > < weekly mo='TRUE' weekFrequency='5' / > < repeatInstances > 5 < /repeatInstances > < /rule > < /recurrence >`
- D. `< recurrence > < rule > < firstDayOfWeek > su < /firstDayOfWeek > < weekly tu='TRUE' weekFrequency='1' / > < repeatInstances > 5 < /repeatInstances > < /rule > < /recurrence >`

**Answer: A**

**Question No : 8 - (Topic 0)**

You create a Microsoft Windows SharePoint Services site that contains a document library.

You need to check out the current version of each document in the root folder of the document

library.

Which code segment should you use?

**A.** `static void CheckOutFiles(SPWeb web, string docLibraryName)`

```
{  
SPListCollection lists = web.Lists;  
lists.IncludeRootFolder = true;  
SPDocumentLibrary docLib =  
(SPDocumentLibrary)lists[docLibraryName];  
foreach (SPFile file in docLib.RootFolder.Files)  
{  
if (file.CheckOutStatus == SPFile.SPCheckOutStatus.None)  
file.CheckOut();  
}  
}
```

**B.** `static void CheckOutFiles(SPWeb web, string docLibraryName)`

```
{  
web.Lists.IncludeRootFolder = true;  
SPDocumentLibrary docLib =  
(SPDocumentLibrary)web.Lists[docLibraryName];  
foreach (SPFile file in docLib.RootFolder.Files)  
{  
if (file.CheckOutStatus == SPFile.SPCheckOutStatus.None)  
file.CheckOut();  
}  
}
```

**C.** `void CheckOutFiles(SPWeb web, string docLibraryName)`

```
{  
web.Lists.IncludeRootFolder = true;  
SPDocumentLibrary docLib =  
(SPDocumentLibrary)web.Lists[docLibraryName];  
foreach (SPListItem item in docLib.Items)  
{  
if (item["CheckOutStatus"].ToString() ==  
SPFile.SPCheckOutStatus.None.ToString()) item["CheckOut"] = true;  
}  
}
```



```

}.
D. static void CheckOutFiles(SPWeb web, string docLibraryName)
{
SPListCollection lists = web.Lists;
lists.IncludeRootFolder = true;
SPDocumentLibrary docLib =
(SPDocumentLibrary)lists[docLibraryName];
foreach (SPListItem item in docLib.Items)
{
if (item["CheckOutStatus"].ToString() ==
SPFile.SPCheckOutStatus.None.ToString()) item["CheckOut"] = true;
}
}
}

```

**Answer: A**

### Question No : 9 - (Topic 0)

You create a records repository for a Microsoft Windows SharePoint Services farm. You need to

enable the Send To menu for a SharePoint site within the farm.

Which code segment should you use?

```

A. protected void RecordsLocation() {
SPWeb app = new SPSite("http: //portal").OpenWeb();
app.Files.Web.GetObject
("http: //portal/records/_vti_bin/officialfile.aspx");
app.Description = "Records";
app.Update();
}

```

```

B. protected void RecordsLocation() {
SPWeb app = new SPSite("http://portal").OpenWeb();
app.CustomMasterUrl =
"http: //portal/records/_vti_bin/officialfile.aspx";
app.Description = "Records";
app.Update();
}

```

```

C. protected void RecordsLocation() {
SPAdministrationWebApplication app =
new SPAdministrationWebApplication();
Uri myurl = new
Uri("http: //portal/records/_vti_bin/officialfile.aspx");

```

```
app.OfficialFileUrl = myurl;
app.OfficialFileName = "Records";
app.Update();
}
D. protected void RecordsLocation() {
SPAdministrationWebApplication app = new
SPAdministrationWebApplication();
string myurl = "http: //portal/records/_vti_bin/officialfile.asmx";
app.PublicFolderRootUrl = myurl;
app.PublicFolderRootUrl.Contains("Records");
app.Update();
}
```

**Answer: C**

### Question No : 10 - (Topic 0)

You are creating a Microsoft Windows SharePoint Services application. You write the following

code segment. (Line numbers are included for reference only.)

```
01 public void DownloadDocument ( SPWeb web){
02 SPDocumentLibrary docLib = (SPDocumentLibrary)
03 web.Lists [Shared Documents];
04 const string fileShare = "C:\\D ownload";
05 ...
06 }
```

You need to move each document to the C:\Download folder. You also need to remove each

document from the document library.

Which code segment should you insert at line 05?

```
A. foreach (SPFolder folder in docLib.Folders) {
folder.MoveTo(fileShare);
}
B. foreach (SPListItem item in docLib.Items) {
string targetPath = Path.Combine(fileShare, item.File.Name);
```