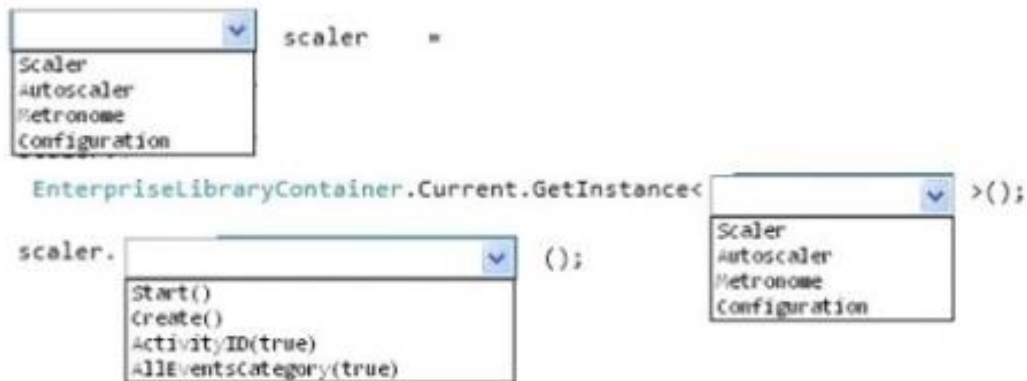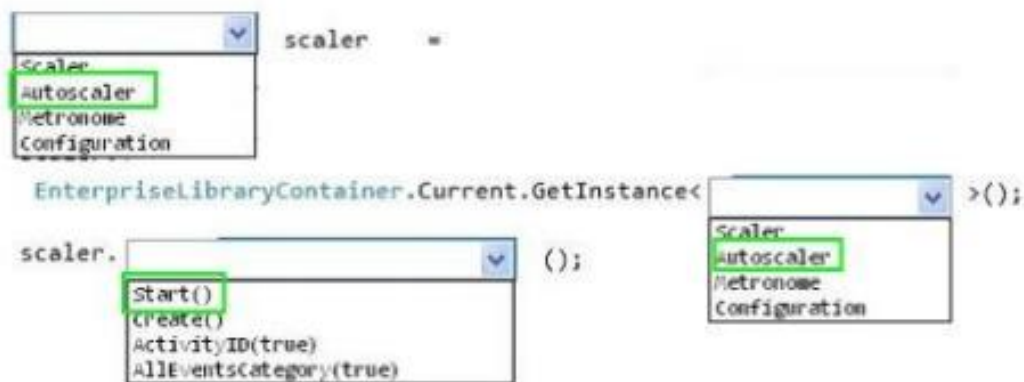1) A company creates an Azure worker role to manage products. The number of customers who inquire about how many products are in inventory rapidly increases. You need to ensure that the worker role can scale to accommodate the increased workload. How should you complete the relevant code?

To answer, select the appropriate option or options in the answer area.

```
[_____▼]   scaler   =
Scaler
Autoscaler
Metronome
Configuration

    EnterpriseLibraryContainer.Current.GetInstance<[_____▼] >();
                                                    Scaler
scaler.[_____▼] ();                      Autoscaler
        Start()                                     Metronome
        Create()                                    Configuration
        ActivityID(true)
        AllEventsCategory(true)
```

Answer:

```
[_____▼]   scaler   =
Scaler
[Autoscaler]
Metronome
Configuration

    EnterpriseLibraryContainer.Current.GetInstance<[_____▼] >();
                                                    Scaler
scaler.[_____▼] ();                      [Autoscaler]
        [Start()]                                   Metronome
        Create()                                    Configuration
        ActivityID(true)
        AllEventsCategory(true)
```

2) You deploy an application as a cloud service in Azure. The application consists of five instances of a web role. You need to move the web role instances to a different subnet. Which file should you update?
   a) Service definition
   b) Diagnostics configuration
   c) Service configuration
   d) Network configuration

   Answer: **C**

3) You have a WebJob object that runs as part of an Azure website. The WebJob object uses features from the Azure SDK for .NET. You use a well-formed but invalid storage key to create the storage account that you pass into the UploadDataToAzureStorage method. The WebJob object contains the following code segment. Line numbers are included for reference only.

```
01 void UploadDataToAzureStorage(CloudStorageAccount storageAccount,
     string storageContainerName, string blobpath, string localpath)
02 {
03    var blobClient = storageAccount.CreateCloudBlobClient();
04    var container = blobClient.GetContainerReference(storageContainerName);
05    CloudBlockBlob blockBlob = container.GetBlockBlobReference(blobpath);
06    blockBlob.UploadFromFile(localpath, FileMode.Open);
07 }
```

For each of the following statements, select Yes if the statement is true. Otherwise, select No.

a) If the storage container does not already exist when the code runs, a file can still be uploaded successfully.

◯ Yes   ◯ No

b) If a transient fault occurs when the code segment on line 06 runs, the Azure SDK will attempt to upload the file again.

◯ Yes   ◯ No

c) The code segment at line 06 will fail when the code runs.

◯ Yes   ◯ No

Answers: **a) No, b) Yes, c) Yes**

4) You have a cloud service that runs an external process that is named MyStartupTask.cmd. The cloud service runs this external process when the web role starts. The external process writes information to the Windows registry. You set the value of an environment variable named MyID to the deployment ID for the current web role instance. The external process must complete writing the information to the Windows registry before the web role starts to accept web traffic. You need to configure the cloud service. How should you complete the relevant markup? To answer, select the appropriate option or options in the answer area.

```
<Startup>
    <Task commandLine="MyStartupTask.cmd"
    [                                                    ]v
    executionContext="elevated" taskType="simple"
    executionContext="limited" taskType="foreground"
    executionContext="elevated" taskType="foreground"
    executionContext="elevated" taskType="background"

        <Environment>
            <Variable name="MyId">
            [                                            ]v
            <RoleInstanceValue xpath="/RoleEnvironment/Deployment/@id"/>
            <RoleInstanceValue xpath="/DeploymentId"/>
            <RoleEnvironment.DeploymentId></Value>
            <Value>@DeploymentId</Value>

            </Variable>
        </Environment>
    </Task>
</Startup>
```

Answer:

```
<Startup>
    <Task commandLine="MyStartupTask.cmd"
```

executionContext="elevated" taskType="simple"
executionContext="limited" taskType="foreground"
executionContext="elevated" taskType="foreground"
executionContext="elevated" taskType="background"

```
        <Environment>
            <Variable name="MyId">
```

<RoleInstanceValue xpath="/RoleEnvironment/Deployment/@id"/>
<RoleInstanceValue xpath="/DeploymentId"/>
<RoleEnvironment.DeploymentId></Value>
<Value>@DeploymentId</Value>

```
            </Variable>
        </Environment>
    </Task>
</Startup>
```

5) Drag and Drop Question. You deploy an application as a cloud service to Azure.
The application contains a web role to convert temperatures between Celsius and Fahrenheit. The application does not correctly convert temperatures.
You must use Microsoft Visual Studio to determine why the application does not correctly convert temperatures. You need to debug the source code in Azure. Which three 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.

Actions	Answer Area
Attach the debugger to the role instance of the cloud service.	
Publish the application.	
In the Microsoft Azure Publish Settings dialog, set the build configuration to **Debug**. and enable the remote debugger for all roles.	
In the Windows Azure Publish Settings dialog, set the build configuration to |**Release**	
In the Microsoft Azure Publish Settings dialog, enable Remote Desktop for cloud configuration and enable the remote debugger for all roles.	

Answer:

Actions

Attach the debugger to the role instance of the cloud service.

Publish the application.

In the Microsoft Azure Publish Settings dialog, set the build configuration to **Debug**. and enable the remote debugger for all roles.

In the Windows Azure Publish Settings dialog, set the build configuration to |**Release**

In the Microsoft Azure Publish Settings dialog, enable Remote Desktop for cloud configuration and enable the remote debugger for all roles.

Answer Area

In the Microsoft Azure Publish Settings dialog, set the build configuration to **Debug**. and enable the remote debugger for all roles.

Publish the application.

Attach the debugger to the role instance of the cloud service.

6) You are designing an upgrade strategy for a Windows Azure application that includes one web role with one instance. You have the following requirements:
- Test the application on the Windows Azure platform.
- Ensure that application upgrades can be rolled back.
- Ensure that upgrade and rollback processes do not cause downtime.
You need to recommend an approach for upgrading the application. What should you recommend?
   a) Deploy to the Production slot.
      Test the application, and then perform a VIP swap.
   b) Deploy to the Staging slot.
      Test the application, and then perform a VIP swap.
   c) Deploy to the Staging slot.
      Test the application, and then perform a manual in-place upgrade to the Production slot.
   d) Deploy to the Staging slot.
      Test the application, and then perform an automatic in-place upgrade to the Production slot.

Answer: B