```
1  public void GetScore(string.playerId, int score, string gameName)
2  {
3     Table Query<DynamicTableEntity> query - new TableQuery<DynamicTableEntity>().Select(new.string[.] { "Score" })
         .Where(TableQuery.GenerateFilterConditionForInt("Score", QueryComparisons.GretaerThanOrEqual, 15000)).Take
(20);
4     EntityResolver<KeyValuePair<string, int?>> resolver =
         (partitionKey, rowKey, ts, props, etag) => new KeyValuePair<string, int?>(rowKey, props["Score"].Int32Value);
5     foreach (var scoreItem in scoreTable.ExecuteQuery (query, resolver, null, null))
6     {
         Console.Writeline($"{scoreItem.Key} {scoreItem.Value}");
7     }
8  }

9  public class PlayerScore : TableEntity
10 {
11    public PlayerScore(string gameId, string playerId, int score, long timePlayed)
12    {
13       PartitionKey = gameId;
14       RowKey = playerId;
15       Score = score;
16       TimePlayed = timePlayed;
17    }
18    public int Score { get; set; }
19    public long TimePlayed { get; set; }
20    }
```

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

You store customer information in an Azure Cosmos database. The following data already exists in the database:

```
01   CloudTableClient tableClient = account.CreateCloudTableClient();
02   CloudTable table = tableClient.GetTableReference("people");
03   TableQuery<CustomerEntity> query = new TableQuery<CustomerEntity>()
04    .Where(TableQuery.CombineFilters(
05   TableQuery.Generate.And, TableQuery.GenerateFilterCondition(Email, QueryComparisons.Equal, "Smith")
06   TableOperstors.And, TableQuery.GenerateFilterCondition(Email, QueryComparisons.Equal,
"ssmith@contoso.com")
07   ));
08   await table.ExecuteQuerySegmentedAsync<CustomerEntity>(query, null);
```

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

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

NOTE: Each correct selection is worth one point.

	Yes	No
The code queries the Azure table and retrieves the TimePlayed property from the table	O	O
The code will display a maximum of twenty records.	O	O
All records will be sent to the client. The client will display records for scores greater than or equal to 15,000.	O	O
The scoreItem.Key property of the KeyValuePairs that ExecuteQuery returns will contain a value for PlayerID.	O	O

Correct Answer:

	Yes	No
The code queries the Azure table and retrieves the TimePlayed property from the table	○	○
The code will display a maximum of twenty records.	○	○
All records will be sent to the client. The client will display records for scores greater than or equal to 15,000.	○	○
The scoreItem.Key property of the KeyValuePairs that ExecuteQuery returns will contain a value for PlayerID.	○	○

QUESTION 27
DRAG DROP
You develop a gateway solution for a public facing news API.

The news API back end is implemented as a RESTful sen/ice and hosted in an Azure App Service instance.

You need to configure back-end authentication for the API Management service instance.

Which target and gateway credential type should you use? To answer, drag the appropriate values to the correct parameters. Each value 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.

NOTE: Each correct selection is worth one point.

Azure Resource
HTTP(s) endpoint
Basic
Client cert

Configuration parameter	Value
Target	value
Gateway credentials	value

Correct Answer:

Azure Resource
HTTP(s) endpoint
Basic
Client cert

Configuration parameter	Value
Target	Azure Resource
Gateway credentials	Client cert

QUESTION 28
You are writing code to create and run an Azure Batch job.

You have created a pool of compute nodes.

You need to choose the right class and its method to submit a batch job to the Batch service.

Which method should you use?

A. JobOperations.CreateJobO
B. CloudJob.Enable(IEnumerable<BatchClientBehavior>)

C.   CloudJob.CommitAsync(IEnumerable<BatchClientBehavior>, CancellationToken)
D.   JobOperations.EnableJob(String, IEnumerable<BatchClientBehavior>)
E.   JobOperations.EnableJobAsync(String. IEnumerable<BatchClientBehavior>. CancellationToken)

Correct Answer: C
Explanation:
A Batch job is a logical grouping of one or more tasks. A job includes settings common to the tasks, such as priority and the pool to run tasks on. The app uses the BatchClient.JobOperations.CreateJob method to create a job on your pool.

The Commit method submits the job to the Batch service. Initially the job has no tasks.

{

CloudJob job = batchClient.JobOperations.CreateJob();

job.Id = JobId;

job.PoolInformation = new PoolInformation { PoolId = PoolId };

job.Commit();

}

References:
https://docs.microsoft.com/en-us/azure/batch/quick-run-dotnet

QUESTION 29
HOTSPOT
A company is developing a Java web app. The web app code is hosted in a GitHub repository located at https://github.com/Contoso/webapp.

The web app must be evaluated before it is moved to production. You must deploy the initial code release to a deployment slot named staging.

You need to create the web app and deploy the code.

How should you complete the commands? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.

```
gitrepo=https://github.com/Contoso/webapp
webappname=businesswebapp
resourcegroupname=BusinessAppResourceGroup
```

az [▼]
- group
- webapp
- appservice plan
- webapp deployment slot
- webapp deployment source

```
create --location centralus - -name $resourcegroupname
create --name $webappname - -resource-group $resourcegroupname
- -sku S3
```

az [▼]
- group
- webapp
- appservice plan
- webapp deployment slot
- webapp deployment source

```
create --name $webappname - -resource-group $resourcegroupname
\ - -plan $webappname
create --name $webappname - -resource-group $resourcegroupname
\ - -slot staging
```

az [▼]
- group
- webapp
- appservice plan
- webapp deployment slot
- webapp deployment source

```
config - -name $webappname - -resource-group $resourcegroupname
\ - -slot staging - -repo-url
$gitrepo - -branch master - -manual-integration
```

az [▼]
- group
- webapp
- appservice plan
- webapp deployment slot
- webapp deployment source

az [▼]
- group
- webapp
- appservice plan
- webapp deployment slot
- webapp deployment source

Correct Answer:

```
gitrepo=https://github.com/Contoso/webapp
webappname=businesswebapp
resourcegroupname=BusinessAppResourceGroup
```

az [▼] `create --location centralus - -name $resourcegroupname`
- group
- webapp
- appservice plan
- webapp deployment slot
- webapp deployment source

`create --name $webappname - -resource-group $resourcegroupname`
`- -sku S3`
`create --name $webappname - -resource-group $resourcegroupname`
`\ - -plan $webappname`

az [▼] `create --name $webappname - -resource-group $resourcegroupname`
- group
- webapp
- appservice plan
- webapp deployment slot
- webapp deployment source

`\ - -slot staging`
`config - -name $webappname - -resource-group $resourcegroupname`
`\ - -slot staging - -repo-url`

az [▼] `$gitrepo - -branch master - -manual-integration`
- group
- webapp
- appservice plan
- webapp deployment slot
- webapp deployment source

az [▼]
- group
- webapp
- appservice plan
- webapp deployment slot
- webapp deployment source

az [▼]
- group
- webapp
- appservice plan
- webapp deployment slot
- webapp deployment source

QUESTION 30
HOTSPOT
You are developing a data storage solution for a social networking app.

The solution requires a mobile app that stores user information using Azure Table Storage.

You need to develop code that can insert multiple sets of user information.

How should you complete the code? To answer, select the appropriate options in the answer area.

NOTE: Each correct selection is worth one point.