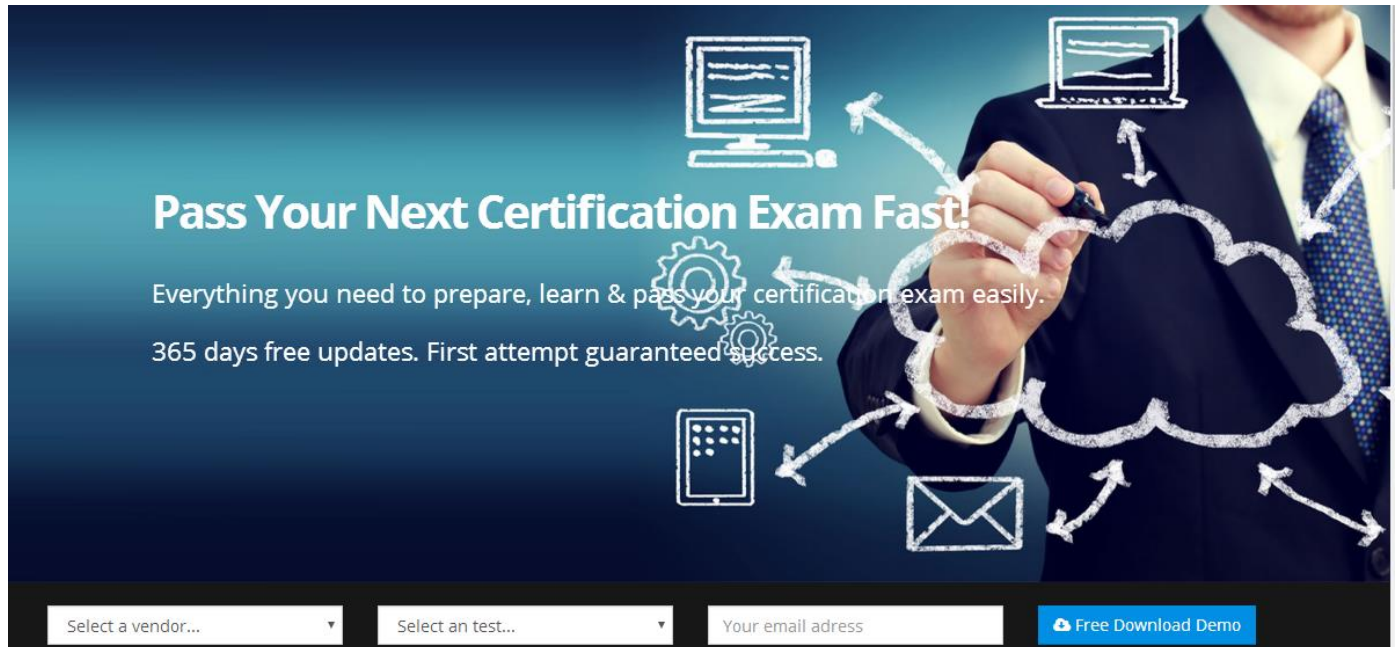


# TestKingfree



**Pass Your Next Certification Exam Fast!**

Everything you need to prepare, learn & pass your certification exam easily.  
365 days free updates. First attempt guaranteed success.

Select a vendor...  Select an test...  Your email address  [Free Download Demo](#)

We're not the only ones **excited** about TestKingFree Practice Material ...

**49625+** customers in 100+ countries use TestKingFree Test Engine. [Meet our customers.](#)

**V VOREED**

**GetCustom**

**JET ORANGE**

**iCompany**

**Paradoxx**


**iMessenger**

## What Client's Say

“ Passed yesterday. Very good valid 300-101 dumps. Only 3-4 questions are new. Most questions and answers are valid. But be careful several answers are incorrect. Study hard. ”

 **Wilbur**  
★★★★★

“ I got 90%. This dumps contains redunant questions and few errors, but defintily enough to pass. :)Prepare well and study much more.Still valid. ”

 **Beatrice**  
★★★★★

<http://www.testkingfree.com/>

Pass For sure Certification Exam Guide and Exam Dumps - TestKingFree

**Exam** : **UiPath-ADAv1**

**Title** : **UiPath Automation Developer Associate v1 Exam**

**Vendor** : **UiPath**

**Version** : **DEMO**

**NO.1** Which of the following activities can be used to get the state of a button (enabled/disabled)?

- A. Find Element
- B. Get Text
- C. Get Attribute
- D. Check App State

**Answer:** C

Explanation:

To determine whether a button is enabled or disabled, the best method is to use the "Get Attribute" activity.

Why is Option C Correct?

\* Get Attribute retrieves various properties of a UI element, including:

\* "enabled" (True/False)

\* "visibility"

\* "aria-disabled"

\* This allows the bot to check if the button is interactable.

Example Use Case:

\* Use Get Attribute activity.

\* Set "Attribute" = "enabled".

\* Store the result in a Boolean variable:

isButtonEnabled = Get Attribute Output

\* Use an If condition to check:

If isButtonEnabled Then

Click Button

Else

Log Message: "Button is disabled"

Why Other Options Are Incorrect?

\* A (Find Element) #

\* Only checks if an element exists but does not retrieve its state.

\* B (Get Text) #

\* Extracts visible text but cannot detect whether a button is enabled/disabled.

\* D (Check App State) #

\* Used to detect application states but not specific element attributes.

**NO.2** What method can be used to change the index of an existing column in a datatable?

- A. SetIndex
- B. MoveAt
- C. SetColumnIndex
- D. SetOrdinal

**Answer:** D

Explanation:

The SetOrdinal method of the DataColumn class can be used to change the index or position of an existing column in a datatable. The method takes an integer argument that specifies the new ordinal or position of the column. For example, to move the first column to the last position, one can use: `yourDataTable.Columns(0).`

`SetOrdinal(yourDataTable.Columns.Count - 1).` (UiPath Automation Developer study guide)

**NO.3** Once "Library A" has been imported as a dependency in the current project, how can the UI Object Repository defined in "Library A" be accessed?

- A.** The Object Repository is only available in a Library.
- B.** The Object Repository needs to be exported as a UI Library and imported in the Process for it to become available.
- C.** The Object Repository will automatically be available in the Process.
- D.** The Object Repository cannot be passed between a Process and a Library.

**Answer:** C

Explanation:

The UI Object Repository is a feature that allows you to store and reuse UI elements across different automation projects<sup>1</sup>. A UI Library is a type of project that contains UI elements and workflows that can be used as dependencies in other projects<sup>2</sup>. When you import a UI Library as a dependency in your current project, the UI Object Repository defined in the UI Library will automatically be available in your current project<sup>3</sup>. You can access the UI elements from the Object Repository panel in Studio and use them in your activities<sup>4</sup>. Therefore, the correct answer is C. The other options are incorrect because they do not reflect the actual behavior of the UI Object Repository and the UI Library.

**NO.4** Which command in the UiPath installation folder configures the UiPath.RemoteDebugging.Agent utility on a Windows robot to accept remote debugging requests from Studio?

- A.** dotnet ./UiPath.RemoteDebugging.Agent.dll enable -- port -- password -- verbose
- B.** UiPath.Remote Debugging.Agent.exe start -- port -- password -- verbose
- C.** UiPath.Remote Debugging.Agent.exe enable -- port -- username -- password -- verbose
- D.** UiPath.Remote Debugging.Agent.exe enable -- port -- password -- verbose

**Answer:** D

Explanation:

The UiPath.RemoteDebugging.Agent utility is a tool that enables remote debugging on a Windows robot<sup>1</sup>. To configure the utility to accept remote debugging requests from Studio, you need to run a command in the UiPath installation folder using the Command Prompt<sup>2</sup>. The command is:

```
UiPath.RemoteDebugging.Agent.exe enable --port <port_number> --password <password> --verbose
```

The arguments in the command are:

\* --port <port\_number>: Specify the port number to use for the remote debugging connection. The default value is 80803.

\* --password <password>: Specify the password to use for the remote debugging connection. The password must match the one entered in Studio when creating a Remote Machine Connection.

\* --verbose: Enable verbose logging for the utility. This argument is optional.

Therefore, option D is the correct command to configure the UiPath.RemoteDebugging.Agent utility on a Windows robot to accept remote debugging requests from Studio. Option A is incorrect because it uses dotnet instead of exe, which is only applicable for macOS robots. Option B is incorrect because it uses start instead of enable, which is only used to start the utility after it is configured. Option C is incorrect because it uses username instead of password, which is not required for the remote debugging

**NO.5** A developer has created an automation that utilizes an Asset value from Orchestrator with the

Get Asset activity. This value represents the email addresses of the process owners, which are subject to change.

What Asset type is most suitable for this situation?

- A. Bool
- B. Integer
- C. Text
- D. Credential

**Answer:** C

Explanation:

The Get Asset activity retrieves stored values from Orchestrator Assets. Since email addresses are text-based values, the most appropriate Asset type is Text.

Why is C Correct?

# Text Asset is suitable for:

- \* Storing strings such as email addresses, URLs, file paths, or API keys.
- \* Easily updated in Orchestrator without modifying the automation.
- \* Supports dynamic values that may change over time.

**NO.6** A developer has opened a pre-existing workflow that utilizes UI Automation within Google Chrome, but the Google Chrome extension has not been installed on their computer.

What steps should the developer take to install the Google Chrome extension from UiPath Studio?

Instructions: Drag the Description found on the left and drop it on the correct Step Sequence found on the right.

**Description**

- Click, "UiPath Extensions".
- Navigate to "Home" and click, "Tools"
- Open Google Chrome and click the "Enable extension" pop-up
- Click the "Install" button for Google Chrome.

**Step Sequence**

Step 1

Step 2

Step 3

Step 4

**Answer:**

**Description**

- Click, "UiPath Extensions".
- Navigate to "Home" and click, "Tools"
- Open Google Chrome and click the "Enable extension" pop-up
- Click the "Install" button for Google Chrome.

**Step Sequence**

- Step 1: Navigate to "Home" and click, "Tools"
- Step 2: Click, "UiPath Extensions".
- Step 3: Click the "Install" button for Google Chrome.
- Step 4: Open Google Chrome and click the "Enable extension" pop-up

Explanation:

Correct Step Sequence:

- \* Navigate to "Home" and click "Tools"
- \* Click "UiPath Extensions"
- \* Click the "Install" button for Google Chrome
- \* Open Google Chrome and click the "Enable extension" pop-up
- \* Step 1: First, navigate to the Home tab in UiPath Studio and click on Tools to access the extension settings.
- \* Step 2: Then, select UiPath Extensions to find available extensions for different browsers.
- \* Step 3: Click on the Install button for Google Chrome to start the installation.
- \* Step 4: Finally, open Google Chrome, and a pop-up will appear asking to enable the extension-this step ensures the extension is active.

**NO.7** At a minimum, what are the recommended properties a developer should configure to use the Get IMAP Mail Messages activity?

- A. MailFolder, Port, Server, Email, and Output Messages.
- B. Account, MailFolder, and Output Messages.
- C. To, Subject, Body, Port, Server, Email, Password or SecurePassword, and Name.
- D. MailFolder, Port, Server, Email, Password or SecurePassword, and Messages.

**Answer:** D

Explanation:

When using the "Get IMAP Mail Messages" activity, the minimum recommended properties to configure are:

MailFolder (to specify the folder from which to retrieve the mails), Port (to specify the port used by the IMAP server), Server (the address of the IMAP server), Email (the email address to connect to), Password or SecurePassword (the password used to connect to the email account), and Messages (the output property where the retrieved emails will be stored).

**NO.8** What are the differences between the Workbook Write Range and Excel Application Integration Write Range activities in writing a DataTable to an Excel file?

- A. (Excel) Write Datatable to Excel activity only works in the background. (Workbook) Write Range activity requires that the file is not opened by the user before running.
- B. (Excel) Write Datatable to Excel activity only works in the foreground, visible to the user.

(Workbook) Write Range activity only works in the background.

**C.** (Excel) Write Datatable to Excel activity will create the file in the WorkbookPath property if it does not exist.

(Workbook) Write Range activity can only be used when Excel is installed.

**D.** (Excel) Write Datatable to Excel activity will execute correctly even if the file is opened by the user before running.

(Workbook) Write Range activity will create the file in the WorkbookPath property if it does not exist.

**Answer:** D

Explanation:

The difference between "Workbook Write Range" and "Excel Application Integration Write Range" activities is that the "Excel Application Integration Write Range" (also known as "Excel Write Range") activity can execute correctly even if the Excel file is already opened by the user before running. It interacts with Excel as an application, making it visible and allowing for interaction with the Excel interface. On the other hand, the "Workbook Write Range" activity works in the background, does not require Excel to be installed, and will create the Excel file in the specified path if it does not exist. So, the correct answer is D.

**NO.9** What is the most appropriate way to convert a designed process into a template after completing the project?

**A.** Use the "Save As" option to save the project as a new file type and manually change the file extension to create a template.

**B.** Use the Export as Template option from the UiPath Studio.

**C.** Create a new Template and move the activities from the existing process to the new Template.

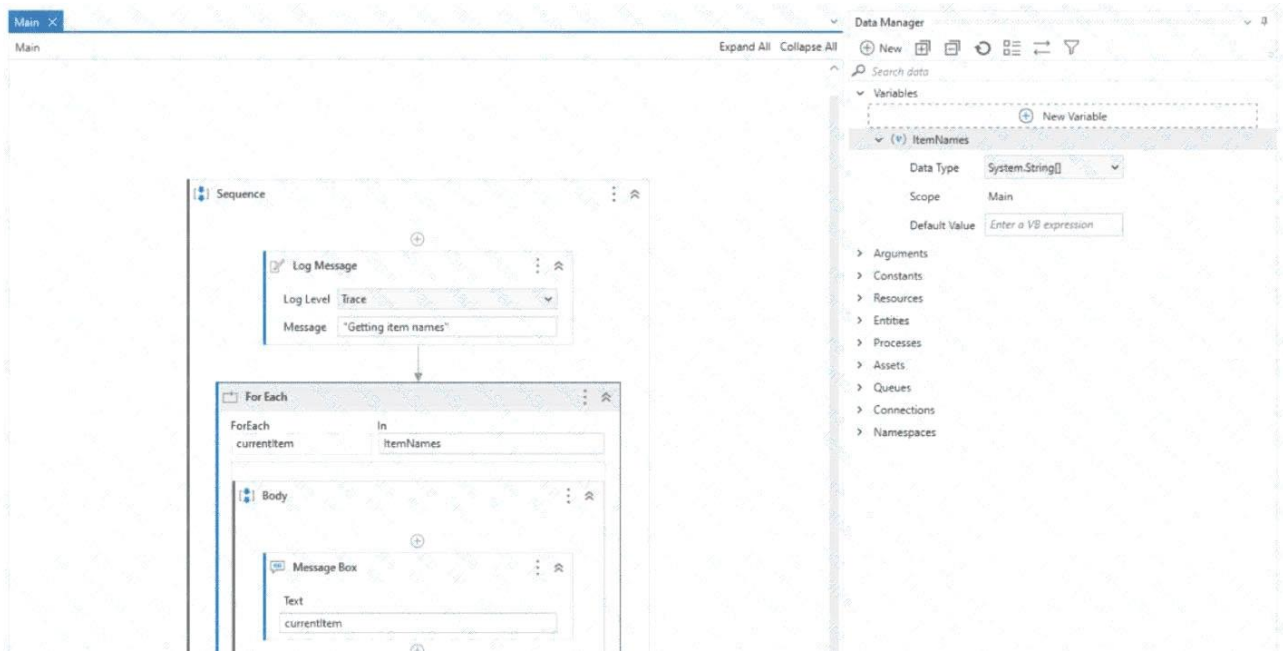
**D.** There is no way the developer can do this.

**Answer:** B

Explanation:

In UiPath Studio, to convert a designed process into a template, you should use the "Export as Template" option. This feature packages the project in such a way that it can be reused as a starting point for other automation projects. Therefore, the correct answer is B.

**NO.10** A developer created a workflow as shown in the following exhibit:



Which exception will be thrown when the workflow is executed?

- A. ArgumentException
- B. InvalidOperationException
- C. BusinessException
- D. NullReferenceException

**Answer: A**

Explanation:

The workflow contains a For Each activity that iterates over the ItemNames variable. However, there is a key issue in the variable definition:

1. Understanding the Issue:

- \* The ItemNames variable is declared as `System.String()`, which is an array of strings.
- \* However, no default value has been assigned to ItemNames. This means the variable is Nothing (null) at runtime.
- \* When the For Each activity attempts to iterate over a Nothing (null) array, an exception is thrown.

2. Why "ArgumentException" is Thrown?

- \* The For Each loop requires a valid enumerable collection to iterate through.
- \* If the collection is Nothing, UiPath will throw an ArgumentException because the input argument (i. e., the collection) is invalid.
- \* The ArgumentException occurs when a method receives an argument that is not valid for its expected purpose.
- \* In this case, the For Each loop expects an initialized array but receives Nothing, leading to an ArgumentException.

3. Why Not Other Options?

- \* B. InvalidOperationException # This is usually thrown when an operation is not valid in the current state, but For Each expects an iterable object and does not throw this error.
- \* C. BusinessException # Business Rule Exceptions are used for process-specific validation errors, which do not apply here.
- \* D. NullReferenceException # This occurs when you try to access a member of an object that is Nothing. However, For Each handles Nothing by throwing an ArgumentException, not a NullReferenceException.

#### 4. Fixing the Issue:

To avoid this error, initialize the ItemNames variable before using it:

vb

CopyEdit

```
ItemNames = New String() {"Apple", "Banana", "Cherry"}
```

Or, check if the array is Nothing before the loop:

vb

CopyEdit

```
If ItemNames IsNot Nothing Then
```

```
For Each currentItem In ItemNames
```

```
' Process each item
```

```
Next
```

```
End If
```


Reference from UiPath Official Documentation:

\* Handling For Each Errors

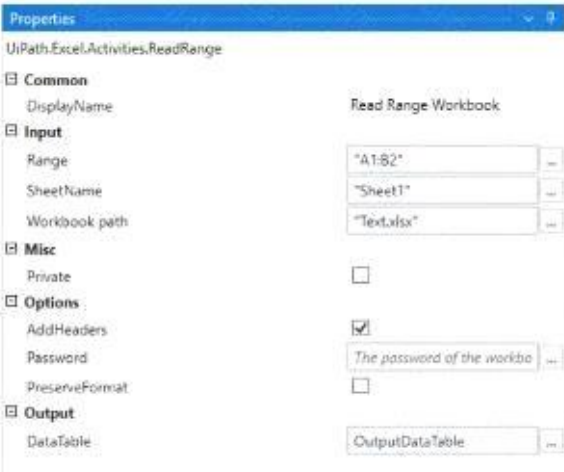
\* Managing Null Values in UiPath

### NO.11

A developer extracts data from "Sheet1" of the "Text.xlsx" file that contains 10,000 rows of data using the activity shown below:



In addition, the developer configured the following properties of the Workbook Read Range activity:



Based on the exhibits, what is the output of the expression `OutputDataTable.Rows.Count`, and what values will be contained in the `OutputDataTable`?

Based on the exhibits, what is the output of the expression `OutputDataTable.Rows.Count`, and what values will be contained in the `OutputDataTable`?

- A. `OutputDataTable.Rows.Count = 1` Contains the values of cells "A2" and "B2"
- B. `OutputDataTable.Rows.Count = 10,000` Contains the values of cells "A2" and "B2"
- C. `OutputDataTable.Rows.Count = 2` Contains the values of cells "A2" and "B2"

**D.** OutputDataTable.Rows.Count = 1 Contains the values of cells "A1" and "B1"

**Answer:** A

Explanation:

Understanding the Read Range Activity Configuration:

\* The Read Range Workbook activity is configured as follows:

\* Range: "A1:B2"

\* Sheet Name: "Sheet1"

\* Headers Option: Unchecked (This means the first row will not be treated as column headers).

\* The Excel file "Text.xlsx" contains 10,000 rows, but the specified range only covers A1 to B2, meaning only two rows of data will be read.

Behavior of Read Range Workbook When "AddHeaders" Is Unchecked:

\* When "AddHeaders" is unchecked, all data in the range is considered part of the data table, including what would normally be treated as column headers.

\* Since the range is "A1:B2", the extracted data includes:

\* Row 1 # A1, B1

\* Row 2 # A2, B2

\* Since the extracted data is treated without headers, the DataTable starts with A1 and B1 as the first row and A2 and B2 as the second row.

Determining OutputDataTable.Rows.Count

\* The Read Range Workbook activity reads both rows (A1:B1 and A2:B2).

\* Since headers are not considered, both rows are stored as data in the DataTable.

\* OutputDataTable.Rows.Count = 2.

# Corrected Output:

OutputDataTable.Rows.Count = 2

DataTable Contains:

Row 1: [A1, B1]

Row 2: [A2, B2]

Why the Correct Answer Is "A" and Not Others:

\* A (OutputDataTable.Rows.Count = 1, Contains A2 and B2) # # Correct if the range was "A2:B2" (Only one row read).

\* B (OutputDataTable.Rows.Count = 10,000) # # Incorrect. The range only selects two rows, not the entire sheet.

\* C (OutputDataTable.Rows.Count = 2, Contains A2 and B2) # # Incorrect. If the row count is 2, the data would include A1 and B1 too.

\* D (OutputDataTable.Rows.Count = 1, Contains A1 and B1) # # Incorrect because two rows were extracted, not one.

**NO.12** How many Global Exception Handlers can be established per automation project, according to best practices?

**A.** A single one for the entire project.

**B.** There is no limit.

**C.** One for each workflow in the project.

**D.** One for each activity in the project.

**Answer:** A

Explanation:

According to best practices in UiPath, there should be a single Global Exception Handler for the entire project, making option A correct. The Global Exception Handler is designed to manage unhandled exceptions in a centralized manner, allowing for a consistent approach to error handling across the entire project. Options B, C, and D suggest multiple handlers, which contradicts the principle of centralized exception management.

**NO.13** In the Output property of all Get Mail activities, what is the supported variable type?

- A. String
- B. List<String>
- C. MailMessage
- D. List<MailMessage>

**Answer:** D

Explanation:

In the Output property of all Get Mail activities, the supported variable type is List<MailMessage>. The Output property is the property that stores the result of the activity in a variable. The Get Mail activities are the activities that retrieve email messages from various mail servers, such as POP3, IMAP, Outlook, or Exchange. The result of the Get Mail activities is a collection of email messages that match the specified criteria, such as the mail folder, the filter, or the top count. The collection of email messages is represented by the List<MailMessage> type, which is a generic list that contains MailMessage objects. A MailMessage object is a class that represents an email message, with properties such as Subject, Body, Attachments, From, To, etc1. For example, if the Get Mail activity is used to retrieve the email messages from the Inbox folder of an Outlook account, the Output property can be configured with a variable of type List<MailMessage>, such as mailList. The mailList variable will store the email messages as a list of MailMessage objects, which can then be accessed or manipulated using other activities, such as For Each, Save Attachments, or Send Mail2.

**NO.14** Consider testing a workflow that computes the sum of two numbers having the data driven test data from the Excel file below:

Number1	Number2	Sum
1	2	3
2	2	4
5	4	9

Expanding the functionality of the workflow to compute the sum of three numbers, the data needs to be updated as well to accommodate the new scenario:

Number1	Number2	Number3	Sum
1	2	2	5
2	2	3	7
5	4	1	10

What are steps to do that?

- A. Click Right on the Test Case and select Update Test Data.
- B. Click Right on the Test Case and select Add Test Data.
- C. Click Right on the Test Case and select Remove Test Data.
- D. Click Right on the Test Case and select Refresh Test Data.

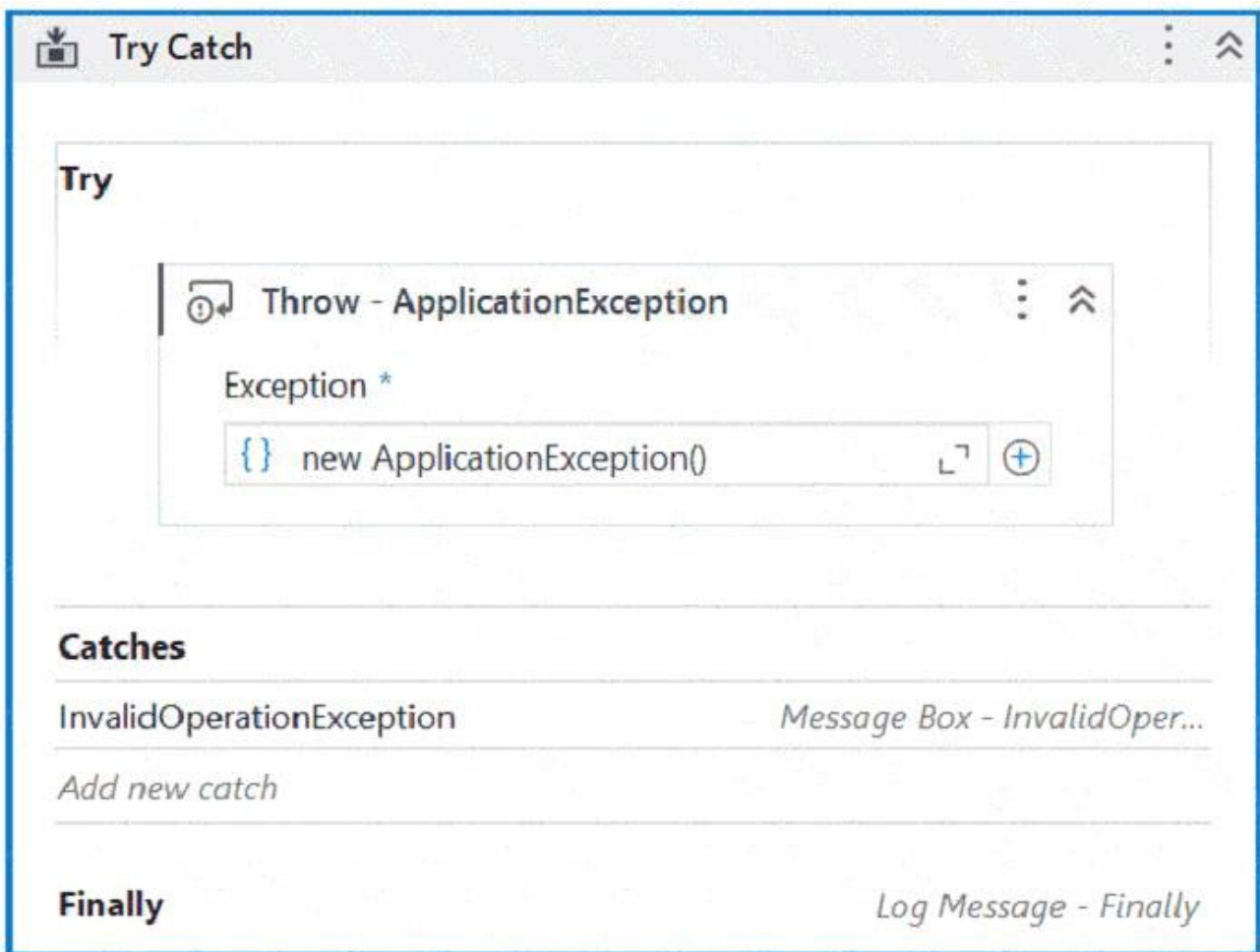
**Answer:** A

Explanation:

To update the data to accommodate the new scenario, you need to modify the Excel file that contains the test data and then refresh the test case to reflect the changes. The steps are as follows:

- \* Open the Excel file that has the test data for the workflow. It should have four columns: Number1, Number2, Sum, and Result.
  - \* Add a new column after Number2 and name it Number3. This column will store the third number for the sum operation.
  - \* Update the existing rows in the Excel file to include a value for Number3 and adjust the value of Sum accordingly. For example, if Number1 is 1, Number2 is 2, and Number3 is 3, then Sum should be 6.
  - \* Save and close the Excel file.
  - \* In UiPath Studio, right click on the test case that uses the Excel file as the test data source and select Update Test Data. This will open a window where you can edit the test data settings.
  - \* In the Update Test Data window, make sure that the Excel file path is correct and that the sheet name is selected. You can also change the name and description of the test data if you want.
  - \* Click OK to save the changes and close the window.
  - \* Right click on the test case again and select Refresh Test Data. This will reload the test data from the Excel file and update the test case accordingly.
- You have successfully updated the test data to compute the sum of three numbers instead of two. You can now run or debug your test case with data variations1.

**NO.15** When encountering an ApplicationException, what occurs if the developer chooses InvalidOperationException as the exception handler within the Catches section of the Try Catch activity?



- A. The Finally block is executed and the Catches section catches the exception.
- B. A runtime error occurs and the Finally block is not executed.
- C. No exceptions are happening and the workflow continues to execute.
- D. No exception is thrown and the Finally block executes.

**Answer:** B

Explanation:

The Try Catch activity is used to handle errors and exceptions that may occur during the execution of a workflow<sup>1</sup>. It has three sections: Try, Catches, and Finally<sup>1</sup>.

\* The Try section contains the activities that may throw an exception or an error. If an exception or an error occurs, the execution of the Try section is stopped and the control is passed to the Catches section<sup>1</sup>.

\* The Catches section contains one or more exception handlers that specify what type of exception or error to catch and what actions to perform when it is caught. The exception handlers are executed in order, from top to bottom, until a matching exception or error is found. If no matching exception or error is found, the execution of the workflow is stopped and a runtime error is thrown<sup>1</sup>.

\* The Finally section contains the activities that are always executed at the end of the Try Catch activity, regardless of whether an exception or error occurred or not. The Finally section is used to perform cleanup actions, such as closing applications, releasing resources, or logging messages<sup>1</sup>.

In your case, you have configured the properties for the Try Catch activity as follows:

\* The Try section contains a Throw activity with an ApplicationException.

\* The Catches section contains an InvalidOperationException with a Message Box activity and a Log

Message activity.

\* The Finally section is empty.

This means that the Try Catch activity will throw an ApplicationException in the Try section and look for a matching exception handler in the Catches section. However, since you have chosen InvalidOperationException as the exception handler, which does not match the ApplicationException, the execution of the workflow will be stopped and a runtime error will occur. The Finally section will not be executed.

Therefore, option B is correct.

**NO.16** Where in the REFramework template project is the "SetTransactionStatus.xaml" invoked?

**A.** In the Finally section of the Try Catch activity in the Process Transaction state.

**B.** In the Try section of the Try Catch activity in the End Process state.

**C.** In the Try and Catches sections of the Try Catch activity in the Process Transaction state.

**D.** In the Catches section of the Try Catch activity in the Process Transaction state.

**Answer:** A

Explanation:

The SetTransactionStatus.xaml file is invoked in the Finally section of the Try Catch activity in the Process Transaction state of the REFramework template project. This file is responsible for setting the status of the current transaction item based on the outcome of the process. It also closes all applications that are used by the process and logs relevant information. (UiPath Studio - REFramework - UiPath Academy)

**NO.17** Which database operations can be controlled with Application Access? (Choose 2 answers)

**A.** Create

**B.** Execute

**C.** Update

**D.** Query

**Answer:** C D

Explanation:

Comprehensive and Detailed Explanation:

In ServiceNow, Application Access settings determine which operations external applications can perform on a table within a Scoped Application.

The primary database operations that can be controlled through Application Access settings are:

# Update (Option C - Correct)

# Query (Option D - Correct)

\* Option A (Create) (Incorrect):

\* The Application Access settings do not explicitly control record creation. Create permissions are typically managed through Access Control Lists (ACLs) rather than Application Access settings.

\* Option B (Execute) (Incorrect):

\* "Execute" is not a valid database operation for Application Access. Execution privileges typically apply to scripted APIs or background scripts rather than database operations.

Example: Controlling Application Access on a Table

In the Table Configuration, under Application Access, you can enable or disable:

# Allow access to this table from other applications

# Can read (Query)

# Can update

# Can delete

These settings restrict or allow other applications to perform query and update operations on the table's records.

**NO.18** When is the Anchor Base activity used to automate interaction with an application?

- A.** When the element selector is not reliable and the element's position on the screen changes
- B.** When the element selector is reliable but the element's position on the screen changes
- C.** When all the element selectors in the application are reliable
- D.** When no element selectors are available in the automated application

**Answer:** A

Explanation:

The Anchor Base activity is used to automate interaction with an application when the element selector is not reliable and the element's position on the screen changes. The Anchor Base activity consists of two parts: an anchor and an action. The anchor is a UI element that has a reliable selector and a fixed position relative to the target element. The action is the activity that performs the desired operation on the target element. The Anchor Base activity uses the anchor to locate the target element based on its relative position (left, right, top, bottom) and then executes the action on it. This way, the automation can handle dynamic and complex UI elements that have unreliable selectors or change their position on the screen. References: Anchor Base and Example for Anchor Base Activity from UiPath documentation and forum.

**NO.19** What is the function of the Immediate Panel in UiPath Studio during the debugging process?

- A.** Inspecting data at a certain point during debugging by evaluating variables, arguments, or statements.
- B.** Displaying the next activity to be executed and its parent containers when the project is paused in debugging.
- C.** Displaying the performance analysis of all the operations, showing the execution time of each activity.
- D.** Displaying the execution process and allowing the modification of variable and argument values.

**Answer:** A

Explanation:

The Immediate Panel in UiPath Studio is primarily used during debugging to evaluate expressions, variables, and arguments in real time.

- \* It allows developers to inspect values dynamically without needing to stop execution.
- \* Useful for debugging complex expressions and checking the result of statements without modifying the workflow.
- \* Works similarly to debugging consoles in development environments like Visual Studio.

# Reference: UiPath Official Documentation - Debugging in Studio

**NO.20** A developer needs to create an array of folder names from the String variable. FilePath = "C:\\Users\\UiPathRobot\\Documents\\Technologies". Based on best practice, which method will return an array of only the folder names?

- A.** Split(FilePath/V)
- B.** Spht(FilePath,"\\": StnngSplitOptions.RemoveEmptyEntries)

- C. SplitV, FilePath, StringSplitOptions.RemoveEmptyEntries)
- D. SplitV', FilePath)

**Answer:** B

Explanation:

To create an array of folder names from the String variable FilePath, the method that will return an array of only the folder names is:

`Split(FilePath, "").StringSplitOptions.RemoveEmptyEntries`

This method uses the `String.Split` method, which returns an array of strings that are separated by a specified delimiter. The delimiter in this case is the backslash character ("`\`") which is used to separate the folder names in the file path. The `StringSplitOptions.RemoveEmptyEntries` option is used to remove any empty entries from the resulting array, such as the one before the first backslash or after the last backslash<sup>1</sup>. For example, if the `FilePath` variable has the value:

`FilePath = "C:\Users\UiPathRobot\Documents\Technologies"`

Then the method `Split(FilePath, "").StringSplitOptions.RemoveEmptyEntries` will return an array of four strings:

```
{"C", "Users", "UiPathRobot", "Documents", "Technologies"}
```

These are the folder names in the file path, without any empty entries.

**NO.21** What is enabled by connecting UiPath Studio to Orchestrator?

- A. Publishing automation packages to Orchestrator.
- B. Defining queues and arguments for use in automations.
- C. Generating API keys for use in integrations with third party apps.
- D. Creating an Object Repository of elements for use in automations.

**Answer:** A

Explanation:

Connecting UiPath Studio to Orchestrator enables several features, including publishing automation packages directly to Orchestrator. This connection is crucial for deploying processes and managing automation workflows at scale. It does not directly enable defining queues and arguments for automations (option B), generating API keys for third-party integrations (option C), or creating an Object Repository of elements (option D), although these can be managed within Orchestrator once the package is published.

**NO.22** Upon completing the development phase, a developer notices that during the User Acceptance Testing, a selector associated with a UI element in the application has been altered. This disruption causes the automation to malfunction and not perform as intended.

According to UiPath guidelines, what is the proper series of actions a developer should follow in order to debug and resolve this issue?

Instructions: Drag the Description found on the left and drop on the correct Step found on the right.

Description		Order of Steps
Examine the error logs to determine the error's source and identify the activity that threw the exception.	Execute the process in Debug mode to confirm that the exception does not occur during runtime.	Step 1
Modify the problematic selector to enhance its flexibility and resilience.	Verify the selector's accuracy, then insert a breakpoint at the activity in question.	Step 2
		Step 3
		Step 4

**Answer:**

Description		Order of Steps
Examine the error logs to determine the error's source and identify the activity that threw the exception.	Execute the process in Debug mode to confirm that the exception does not occur during runtime.	Step 1
Modify the problematic selector to enhance its flexibility and resilience.	Verify the selector's accuracy, then insert a breakpoint at the activity in question.	Step 2
		Step 3
		Step 4

**Explanation:**

Description (Left)	Correct Order of Steps (Right)
Examine the error logs to determine the error's source and identify the activity that threw the exception.	Step 1
Verify the selector's accuracy, then insert a breakpoint at the activity in question.	Step 2
Modify the problematic selector to enhance its flexibility and resilience.	Step 3
Execute the process in Debug mode to confirm that the exception does not occur during runtime.	Step 4

When a UI element selector changes, it can break automation. The proper debugging sequence ensures an effective fix while preventing future failures.

Step 1: Examine Error Logs

# First, review error logs in the Output Panel to locate the failing activity.

- \* This helps determine which selector is causing the issue.
- \* Use UiPath's Exception Details to pinpoint the problem.

Step 2: Verify Selector and Insert Breakpoint

# Open UI Explorer to inspect the failing selector.

- \* Compare the new selector vs. old selector.
- \* Use a Breakpoint at the failing activity to analyze execution in Debug Mode.

Step 3: Modify Selector for Flexibility

# Adjust the selector to make it more resilient by:

- \* Using Wildcards (\*) for dynamic attributes.

- \* Implementing Anchor-based selectors.
- \* Utilizing Modern UI Automation if applicable.

Step 4: Execute in Debug Mode

# Run the automation in Debug Mode to verify that:

- \* The updated selector correctly identifies the UI element.
- \* No further exceptions occur during runtime.

**NO.23** The "Main.xaml" file has an "In" argument of type String with a default value of "String.Empty". What is the runtime value of the argument when executing the process from the Start Job window in UiPath Orchestrator?

- A.** A runtime error will occur as arguments are not allowed in the "Main.xaml" file.
- B.** Only the default value of "String.Empty" will be used, overriding any values provided in Orchestrator.
- C.** @ If any input arguments are provided in Orchestrator, those values will be used; otherwise, the default value of "String.Empty" will be used.
- D.** An Input Dialog box will prompt the user to input a value and that value will be used.

**Answer:** C

Explanation:

When a process is executed from the Start Job window in UiPath Orchestrator, the runtime value of an "In" argument in the "Main.xaml" file will be the value provided in Orchestrator if specified. If no value is provided in Orchestrator for this argument, then the default value specified in the "Main.xaml" file will be used, in this case, "String.Empty". Therefore, option C is correct. Arguments can be passed to the "Main.

xaml" file from Orchestrator, and there is a mechanism to handle default values for arguments in case they are not provided at runtime.

**NO.24** What happens when closing a Remote Debugging Connection while a debugging execution is in progress?

- A.** Debugging execution stops gracefully, then the connection is closed.
- B.** The remote robot continues the execution after Studio closes the connection.
- C.** It is not possible to close the connection while debugging is in progress.
- D.** Debugging execution ends in exception, then the connection is closed.

**Answer:** D

Explanation:

When you close a Remote Debugging Connection while a debugging execution is in progress, the execution is stopped abruptly and an exception is thrown on the remote robot. The connection is then closed and you are returned to the Studio interface. (Debugging in Studio - UiPath Academy)

**NO.25** What is a pre-requisite for running functional test cases in REFramework?

- A.** Invoke Process XAML file
- B.** Invoke InitAllSettings XAML file
- C.** Invoke SetTransactionStatus XAML file
- D.** Invoke Main XAML file

**Answer:** B

Explanation:

A pre-requisite for running functional test cases in REFramework is to invoke the InitAllSettings XAML file, which initializes the application, reads the configuration file, and sets the log level. This file is invoked in the Init state of the Main XAML file, which is the entry point of the REFramework. By invoking the InitAllSettings file, you can ensure that the test cases have access to the necessary settings and parameters for the automation project. (UiPath Studio - Testing Frameworks - UiPath Academy)

**NO.26** When creating a new project in Studio Web with cross-platform compatibility, what language must be selected for expressions to ensure the project is cloud-compatible?

- A. VB
- B. C#
- C. Java
- D. Python

**Answer:** B

Explanation:

Studio Web is UiPath's cloud-based automation platform that enables cross-platform compatibility (Windows, Linux, Mac).

Why is B Correct?

# C# is the recommended language because:

- \* Cloud projects in UiPath are built using .NET Core, which supports C# natively.
- \* VB is not supported in cross-platform execution.
- \* C# ensures compatibility with Linux-based cloud environments.

**NO.27** The provided exhibit displays the representation of the "Deadline" field using the yyyy-mm-dd format.

Queue Item Name	Deadline	Priority
EmployeeID1		Normal
EmployeeID2	2050-01-01	High
EmployeeID3	2050-01-02	High
EmployeeID4	2050-01-01	Normal
EmployeeID5	2050-01-02	Low
EmployeeID6		High
EmployeeID7	2050-01-01	Low
EmployeeID8	2050-01-01	Normal
EmployeeID9		Low

In what order should the Queue Items be processed by a single robot executing a Performer workflow?

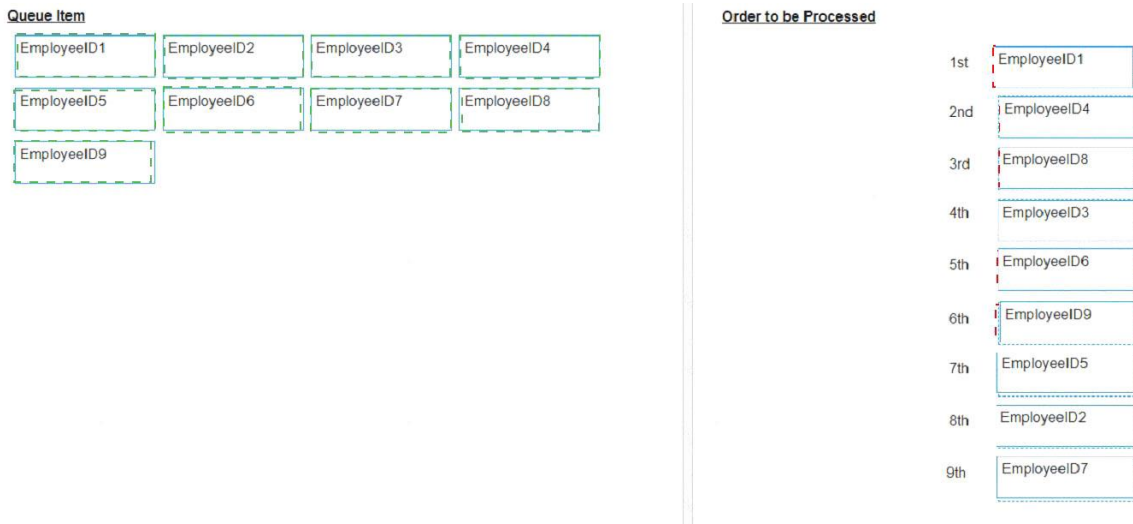
Queue Item

EmployeeID1	EmployeeID2	EmployeeID3	EmployeeID4
EmployeeID5	EmployeeID6	EmployeeID7	EmployeeID8
EmployeeID9			

Order to be Processed

1st	<input type="text"/>
2nd	<input type="text"/>
3rd	<input type="text"/>
4th	<input type="text"/>
5th	<input type="text"/>
6th	<input type="text"/>
7th	<input type="text"/>
8th	<input type="text"/>
9th	<input type="text"/>

**Answer:**



**Explanation:**

The Queue Items should be processed by a single robot executing a Performer workflow in the following order:

- \* EmployeeID1
- \* EmployeeID4
- \* EmployeeID8
- \* EmployeeID3
- \* EmployeeID6
- \* EmployeeID9
- \* EmployeeID5
- \* EmployeeID2
- \* EmployeeID7

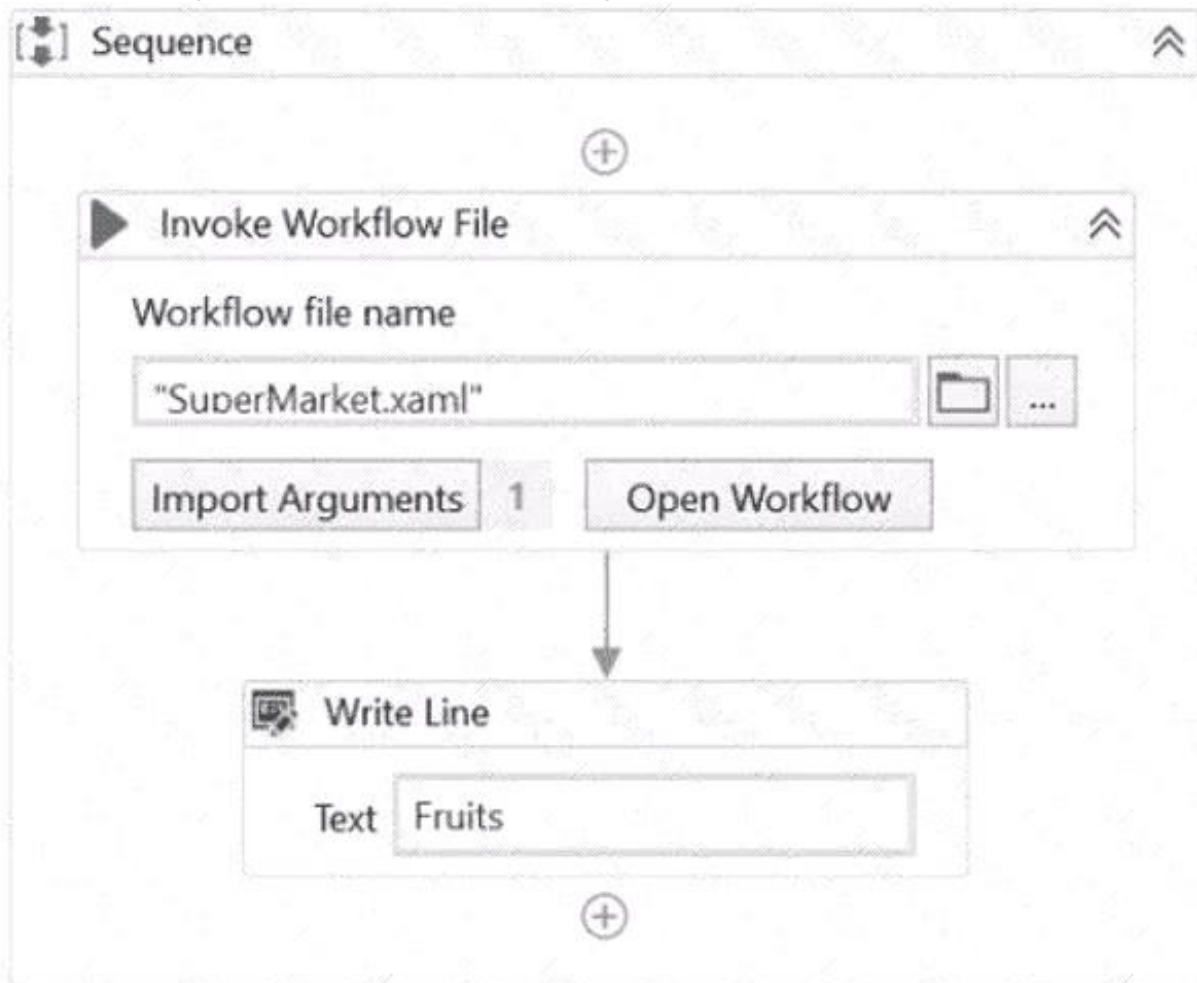
A queue is a container that holds an unlimited number of items that can be processed by a robot<sup>1</sup>. A queue item can have a deadline and a priority, which affect the order in which the robot retrieves the items from the queue<sup>2</sup>. The Performer workflow is a part of the REFramework template that processes the queue items using the Get Transaction Item activity<sup>3</sup>. The order of queue execution is determined by the following rules<sup>4</sup>:

- \* Items that have a deadline are processed first, in order of priority and according to the set deadline for items with the same priority.
- \* Items that have no deadline are processed next, in order of priority and according to the rule First In, First Out for items with the same priority.

Based on these rules and the exhibit, the order of queue execution is as follows:

- \* EmployeeID1 has the earliest deadline (2050-01-01) and high priority, so it is processed first.
- \* EmployeeID4 has the same deadline as EmployeeID1, but normal priority, so it is processed second.
- \* EmployeeID8 has the same deadline and priority as EmployeeID4, but it was added later to the queue, so it is processed third.
- \* EmployeeID3 has the next earliest deadline (2050-01-02) and high priority, so it is processed fourth.
- \* EmployeeID6 has the same deadline as EmployeeID3, but normal priority, so it is processed fifth.
- \* EmployeeID9 has the same deadline and priority as EmployeeID6, but it was added later to the queue, so it is processed sixth.
- \* EmployeeID5 has no deadline, but high priority, so it is processed seventh.
- \* EmployeeID2 has no deadline and normal priority, so it is processed eighth.
- \* EmployeeID7 has no deadline and low priority, so it is processed ninth.

**NO.28** A developer has created an automation process that includes:



2. Variables in the Variable panel in the Main.xaml file:

Name	Variable type	Scope	Default
Fruits	String	Sequence	"Apple"

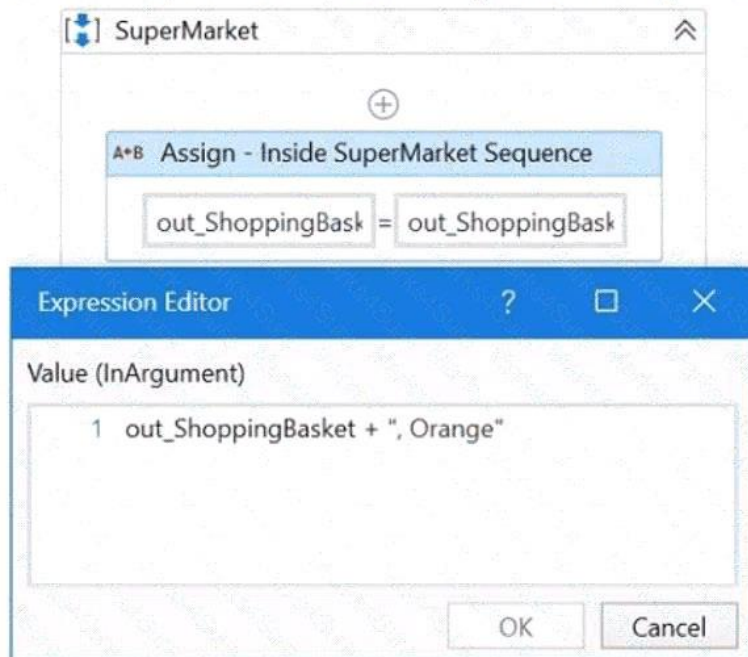
Create Variable

3. Main.xaml file invokes the SuperMarket workflow using the following argument:

Name	Direction	Type	Value
out_ShoppingBasket	Out	String	Fruits

Create Argument

4. Detailed SuperMarket sequence and right-hand side expression shown below:



Based on the information shown in the exhibits, what is the output of the Write Line activity in the Main xaml file?

- A. Apple
- B. Orange
- C. Apple Orange
- D. Orange Apple

**Answer:** B

Explanation:

The Write Line activity is used to write a specified text to the Output panel. The text can be a string literal, a variable, or an expression. In this case, the Write Line activity in the Main xaml file has the text

"out\_ShoppingBasket + ", "Orange"". The out\_ShoppingBasket is a variable of type String, which is declared in the Main xaml file and has the default value of "Apple". The variable is passed as an argument to the SuperMarket xaml file, where it is assigned a new value of "Orange". The argument direction is Out, which means that the value of the argument is passed back to the Main xaml file after the execution of the SuperMarket xaml file. Therefore, the value of the out\_ShoppingBasket variable in the Main xaml file is changed from "Apple" to "Orange". The expression "out\_ShoppingBasket + ", "Orange"" concatenates the value of the out\_ShoppingBasket variable with a comma and a space, followed by the string literal "Orange".

The result of this expression is "Orange, Orange". The Write Line activity writes this text to the Output panel.

Therefore, the answer is B. Orange. References: Write Line, Variables, Arguments

**NO.29** To read only the first page of a .pdf file using the Read PDF Text activity, what value should be entered into the Range property?

- A. "1"
- B. ""

C. (0)

D. 1

**Answer:** A

Explanation:

The Read PDF Text activity in UiPath allows users to extract text from a PDF file. The Range property determines which pages should be read.

\* Correct Syntax for Reading a Specific Page:

\* To read only page 1, you need to enter "1" in the Range property.

\* This tells UiPath to extract text only from the first page of the document.

\* Why Other Options Are Incorrect:

\* B ("") # An empty string means all pages will be read, which is not what the question asks.

\* C ((0)) # Incorrect because UiPath follows 1-based indexing, not 0-based.

\* D (1) # Incorrect because the Range property expects a string value ("1"), not a numeric input.

**NO.30** What role do Triggers play in the UiPath Integration Service?

**A.** Manage connections between UiPath Studio and third-party applications.

**B.** Provide a mechanism for starting processes on a scheduled basis from Orchestrator.

**C.** Assist in the creation of automation projects by providing event-based activities.

**D.** Provide a mechanism for subscribing to specific events from third-party applications, automatically starting processes in Orchestrator.

**Answer:** D

Explanation:

Triggers are a feature of UiPath Integration Service that allow you to react to key events in third-party systems, such as data updates, insertions, deletions, or custom events. By configuring triggers, you can start your UiPath processes automatically from Orchestrator when these events occur, enabling near real-time automation workflows across different applications. (UiPath Integration Service documentation1)

**NO.31** You want to automate document processing tasks such as extracting and validating data from multiple documents in an online service. How can you achieve this using UiPath Studio Web?

**A.** Use the Document Understanding feature in Studio Web to extract, process, and validate documents.

**B.** Rely on the native data export functions of the online service to automate document processing without using Studio Web.

**C.** Automate manual data entry by creating a loop in Studio Web to input document details into a spreadsheet.

**D.** Build a custom script in Studio Web using programming code to extract and validate data from documents.

**Answer:** A

Explanation:

UiPath Studio Web supports Document Understanding for processing structured and unstructured documents in cloud-based automation.

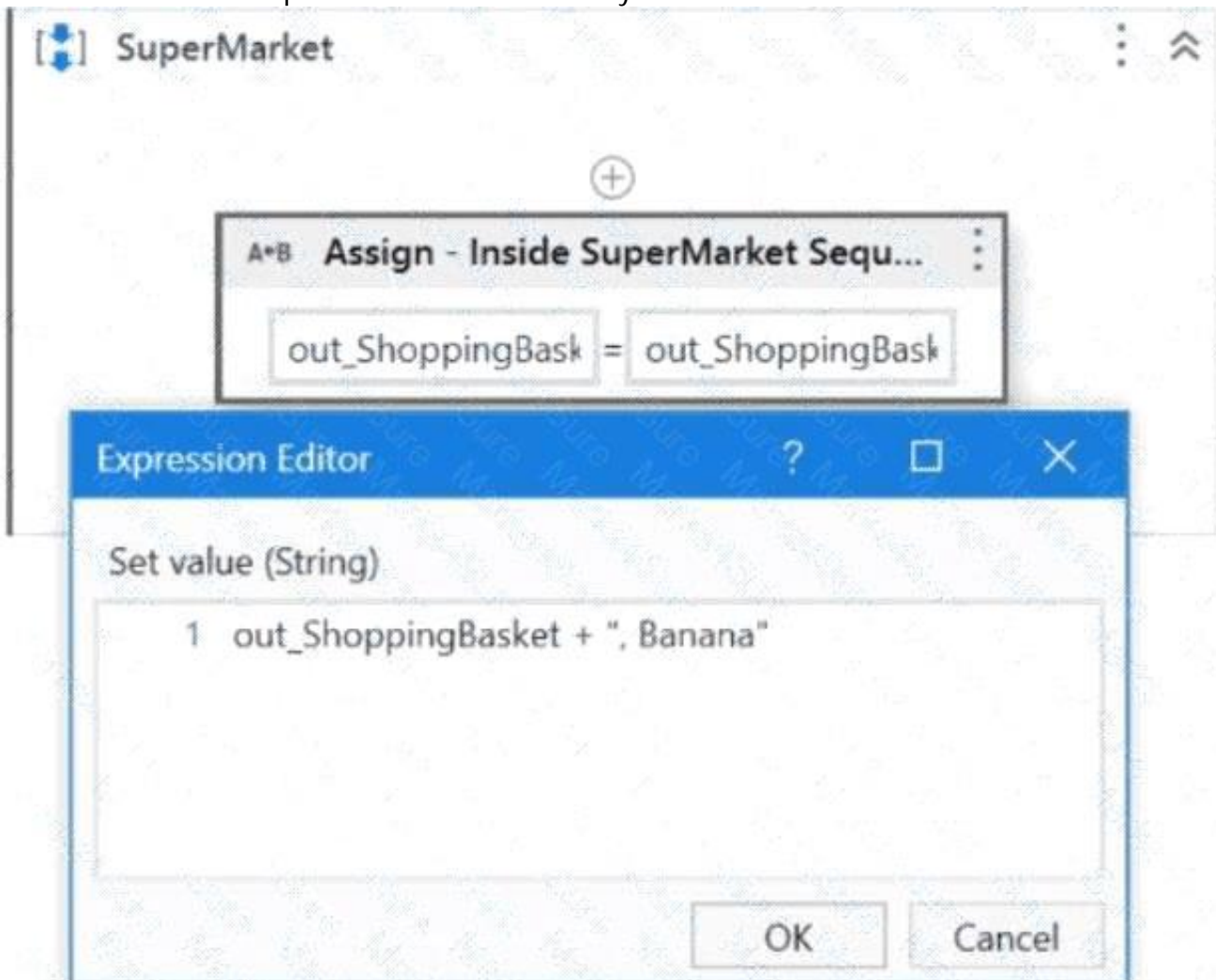
\* The Document Understanding feature in Studio Web allows:

\* Extracting key data fields from documents.

\* Validating and classifying document types.

- \* Using AI models and OCR engines for text extraction.
- \* This is the most efficient way to handle large-scale document processing without manual intervention.
- # Reference: UiPath Studio Web - Document Understanding

**NO.32** What is the output of the Write Line activity in the "Main.xaml" file?



- A. , Banana
- B. Banana
- C. Watermelon
- D. Watermelon, Banana

**Answer:** D

Explanation:

Comprehensive and Detailed In-Depth Explanation:

This question is based on passing arguments between workflows in UiPath.

Step-by-Step Execution Guide: Passing Arguments Between Workflows

1## Main.xaml contains:

\* A variable Fruits = "Watermelon".

2## SuperMarket.xaml contains:

\* An argument out\_ShoppingBasket = out\_ShoppingBasket + ", Banana".

3## Execution Flow:

- \* SuperMarket.xaml receives "Watermelon".
- \* The out\_ShoppingBasket variable appends ", Banana" to "Watermelon".
- \* Main.xaml logs the final value: "Watermelon, Banana".
- # This demonstrates how arguments modify data across workflows!
- # Reference:
- \* UiPath Documentation: Working with Arguments

**NO.33** Which LINQ method is used to filter data in a List or Data Table based on a condition?

- A. OrderBy
- B. Where
- C. GroupBy
- D. Select

**Answer:** B

Explanation:

The Where method is used to filter data in a List or Data Table based on a condition. It returns a new collection that contains only the elements that satisfy the condition<sup>12</sup>. The other methods are used for different purposes: OrderBy for sorting, GroupBy for grouping, and Select for projecting<sup>3</sup>.

**NO.34** A developer downloads a published package from UiPath Orchestrator 2021.10 to use locally. What is the extension of the downloaded package?

- A. .csproj
- B. .dll
- C. .nupkg
- D. .zip

**Answer:** C

Explanation:

The extension of the downloaded package from UiPath Orchestrator 2021.10 to use locally is .nupkg. A package is a file that contains the workflows and the dependencies of a project that is ready to be executed by a robot. A package can be created and published from UiPath Studio, or uploaded manually to UiPath Orchestrator. A package can also be downloaded from UiPath Orchestrator to use locally, such as for debugging or testing purposes. The extension of the package file is .nupkg, which is the standard extension for NuGet packages. NuGet is a package manager that is used to manage the dependencies of a project, such as libraries, frameworks, or components<sup>3</sup>. A .nupkg file is essentially a ZIP archive that contains the files and metadata of the package<sup>4</sup>. For example, if a package named Process1 is downloaded from UiPath Orchestrator, the file name and extension will be Process1.nupkg. The .nupkg file can be opened or extracted using a ZIP utility, such as 7-Zip or WinZip, or renamed to .zip and extracted using Windows Explorer<sup>5</sup>.

**NO.35** Which of the following functionalities does UiPath Assistant provide?

- A. Developing automation workflows in UiPath Studio.
- B. Running, managing, and organizing automation workflows on the user's machine.
- C. Scheduling and monitoring robot processes in Orchestrator.
- D. Analyzing processes to determine optimal automation solutions.

**Answer:** B

Explanation:

UiPath Assistant is a desktop tool that enables users to easily access, manage, and run automation workflows on their machines. It provides the following functionalities:

- \* Viewing all the available automations that the user can run, either from Orchestrator or locally.
- \* Adding automations to the launchpad for quick and easy access.
- \* Creating custom folders and sections to organize the automations according to the user's preferences.
- \* Running automations with a single click or using keyboard shortcuts.
- \* Launching robot-powered apps and managing action center tasks directly from the assistant.
- \* Customizing the assistant's appearance, name, and avatar to suit the user's personality.
- \* Viewing the status and progress of the running automations, as well as the history and logs of the past executions.
- \* Connecting to Orchestrator and synchronizing the automations and settings with the cloud.
- \* Accessing the help and feedback options, as well as the interactive guided tour.