Module 3 QB

-PP

1) What is sequence in UiPath? Explain with example.

Ans) A Sequence is a group of logical steps. Each step represents an action or a piece of work.

Example:

1. Add a Sequence activity to the Designer panel.

2. Next, add a For each activity inside the Sequence.

3. Create two variables; an integer variable named item, and an array integer variable named x. Then, set them to the text field.

4. Now, assign a default value to the integer variable x.

5. Add a Break activity inside the body of the loop.

6. Under the For Each activity, add a Write line activity.

7. In the Write line activity, type item.ToString in the text field.

2) What are activities and how can activities be used along with workflows?

Ans)

Activity:

- In UiPath Studio, an activity represents the unit of an action.
- Each activity performs some action.
- When these activities combine together, it becomes a process.
- Every activity resides on the Activities panel of the main Designer panel.

- Suppose you have a big project that consists of hundreds of activities.
- To build such a big project, a developer will simply divide it into smaller modules and extract it as a workflow.
- Now, each workflow can be tested separately.
- Thus, it is very easy to find bugs and errors.
- UiPath provides numerous activities for performing the decision-making process.
- These activities, present in the Activities panel, are put into the workflow either using the double-click method or the drag and drop method.

3) i) List the different operations used in excel file.

Ans) The following are the operations that are frequently used with an Excel file:

- Read cell
- Write cell
- Read range
- Write range
- Append range

ii) Write the steps to build the following bots.

a) Read cell

Ans)

1) Drag and drop a Flowchart activity on the main Designer panel.

2) Also, drag and drop an Excel application scope inside the Flowchart. Connect it to the Start node.

3) Double click on Excel application scope.

4) Drag and drop the Read Cell activity inside the Excel application scope activity.

5) Specify the range value in the cell text box of the Read Cell activity.

6) Create a variable of type string to hold the result produced by the Read Cell activity.

7) In our case, we have created a Result variable.

8) Specify the Output property of the Read Cell activity by providing the variable's name that we have created.

9) Drag and drop a Message box activity inside the Excel application scope activity and specify the string variable's name here.

10) That's it. Press F5 to see the result.

b) Write cell

Ans)

1) Drag and drop a Flowchart activity on the main Designer panel.

2) Also, drag and drop an Excel application scope inside the Flowchart activity. Connect it to the Start node.

3) Drag and drop a Write Cell activity inside the Excel application scope.

4) Specify the cell value in which we want to write in the Range property of the Write Cell activity.

5) Also, specify the value of the Value property.

6) Press F5 and see the result.

7) Open the Excel file to see the changes.

c) Read range

Ans)

1) Drag and drop a Flowchart activity on the main Designer panel.

2) Also, drag and drop an Excel application scope inside the Flowchart activity. Connect it to the Start node.

3) Drag and drop a Read Range activity inside the Excel application scope activity.

4) The Read Range activity produces a data table.

5) We need to create a data table variable and specify it in the Output property of the Read Range activity.

6) Drag and drop an Output Data Table activity inside the Excel application scope activity.

7) Now, we have to specify two properties of the Output Data Table activity: Data Table property and text property.

8) The Data Table property of the Output Data Table activity is used to convert the data table into a string format. The text property is used to supply its value in a string format. We have to receive this value in order to consume it.

9) So let us create a variable of type string and give it a meaningful name.

10) Drag and drop a Message box activity inside the Excel application scope activity.

11) Also, specify the string variable's name that we created earlier inside the Message box activity.

12) Press F5 to see the result.

13) A window will pop up displaying your Excel file data.

d) Write range

Ans)

1) Drag and drop a Build data table activity from the Activities panel.

2) Double-click on this activity. A window will pop up.

3) You will notice that two columns have been generated automatically. Delete these two columns.

4) Add your column by clicking on the + icon and specify the column name. You can also select your preferred data type. You are free to add any number of columns.

5) In this project, we are adding two columns.

6) The procedure for adding the second column is almost the same like first.

7) We have added one more column (Roll) and set the data type to Int32 for the data table. We have also initialized this data table by providing some values in its rows.

8) Create a variable of type data table. Give it a meaningful name.

9) Specify this data table name in the Data table property of the Build data table activity.

10) Our data table has been built successfully.

11) Drag and drop an Excel application scope inside the main Designer panel.

12) Specify the Excel sheet path there.

13) Connect this activity to the Build Data Table activity.

14) Inside the Excel application scope activity, just drag and drop the Write Range activity.

15) Specify the data table variable name that we created earlier and set it as a Data table property inside the Write Range activity. We can also specify the range.

16) In this case, we have assigned it as an empty string.

17) Hit the Run button to see the result.

e) Append range

Ans)

1) Drag and drop the Flowchart activity on the main Designer window.

2) Also, drag and drop the Excel application scope inside the Flowchart activity. Connect it to the Start node.

3) The Append Range activity requires a data table.

4) Drag and drop the Read Range activity inside the Excel application scope activity.

5) The Read Range activity produces a data table.

6) We have to receive this data table in order to consume it. Create a data table variable and specify it in the Output property of the Read Range activity.

7) Drag and drop the Append Range activity inside the Excel application scope activity.

8) Specify the Excel file path in the Append Range activity.

9) Also, specify the data table (which is generated by the Read Range activity).

10) Press F5 to see the result.

4) What is argument? What are the four directions of argument?

Ans) An Argument is simply a variable that can store a value.

The four directions of argument are:

- **In:** When we have to receive the value from another workflow.
- **Out:** If we have to send the current value to a workflow.
- **In/Out:** This specifies both; it can take or receive the value.
- **Property:** This specifies that it is not being used currently.

5)

i) List out the different types of control flow activities.

Ans)

Different types of control flow activities are as follows:

- The Assign activity
- The Delay activity
- The Break activity
- The While activity
- The Do While activity
- The For each activity
- The If activity
- The Switch activity

ii) Explain the following activities with examples.

a) The While activity:

Ans) The While activity is used in automation to execute a statement or process based on a certain condition.

If found true, the loop is executed.

Example: We will see how an integer variable will increase from 5 to 50 in increments of 5.

Perform the following steps:

1. On a Blank project, add a Sequence activity.

2. Now, create an integer type variable Y. Set its default value to 5.

3. Next, add a While activity to the Sequence.

4. In the condition field, set x<50.

5. Add an Assign activity to the body section of the While loop.

6. Now, go to the Properties panel of the Assign activity and type in the text field integer variable for value field integer x+5

7. Drag and drop a Write line activity and type x.ToString in the text field.

b) The Do While activity:

Ans) The Do while activity is used in automation when it is required to execute a statement based on the fulfillment of a certain condition.

Example: Take an integer variable. Starting with this variable, we shall generate all multiples of 2, less than 20. Perform the following steps:

1. Add a Sequence to the Designer panel.

2. Add a Do while activity from the Activities panel.

3. In the body section of the Do while activity, add an Assign activity.

4. Now, select the Assign activity. Go to the Properties panel and create an integer variable y. Set its default value to 2.

5. Set y+2 in the value section of the Assign activity to increment the result each time by until the loop is executed.

6. Add a Write line activity inside the Assign activity.

7. In the text field of the Write line activity, type Z.

8. In the condition section, set the condition y <20. The loop will continue until the condition

holds true.

9. On clicking the Run button, the output for each iteration will be displayed.

c) The For Each activity:

Ans) The For each activity works by iterating each element from the collection of items or list of elements, one at a time.

Example: We shall use the For each activity to go through a collection of even numbers and display each element one at a time.

Perform the following steps:

1. Start with a Blank project in UiPath.

2. Add a Sequence activity to the Designer panel.

3. Next, add a For each activity within the Sequence and create an integer type array variable, Y.

4. In the default value of the variable, put in ({2,4,6,8,10,12,14,16,18,20}).

5. Add a Write line activity to the Designer Panel.

6. In the text field of the Write line activity, type item.ToString to display the output.

7. Now, run the program. You will see that each number of the array is displayed one by one because of the use of the For each activity.

d) If activity:

Ans) The If activity consists of a statement with two conditions: true or false.

- If the statement is true, then the first condition is executed; if not, the second condition is executed.
- Example: Perform the following steps:

1. Add a Flowchart from the Activities panel.

2. Add two Input dialog activities. Create two integer variables, x and y.

3. In the Properties panel, change the label name and title name of both the Input dialog activities.

4. Now, specify these name of these two variables in the Result property of both the Input dialog activities.

5. Now add the If activity to the Designer panel

6. In the condition part, x+y<6, check whether it is true or false. Add two Write line activities and type “True” in one and “False” in the other.

7. Click the Run button to check the output. If the condition holds true then it will show the true value; otherwise, it will show the false value.

e) The Switch Activity:

Ans) The Switch activity can be used to make a choice. When we have various options available and want to execute one option, we frequently use the Switch activity.

- Example: Perform the following steps:

1. Add a Sequence activity.

2. Add an Input dialog activity inside the Sequence.

3. Now, create an integer type variable k.

4. Specify the newly created variable's name in the Result property inside the Properties panel.

5. Add the Switch activity under the Input dialog activity.

6. In the Expression field, set k mod 2 to check whether the number is divisible by 2 or not.

7.Add a Write line activity to the Default section and type the k.ToString +”is an even number” in the text field.

8. Now, create Case 1, add the one other Write line activity to it, and type k.ToString +”is an odd number” in the text field.

f) Assign activity:

Ans) The Assign activity is used to designate a value to the variable.

The Assign activity can be used for different purposes, such as incrementing the value of a variable in a loop, or using the results of a sum, difference, multiplication, or division of variables and assigning it to another variable.

g) The Delay activity:

Ans) The Delay activity is used to delay or slow down an automation by pausing it for a defined period of time.

Example: Let us see an example of an automation that writes two messages to the Output panel with a delay of 50 seconds.

1) First, create a new Flowchart.

2) Add a Write line activity from the Activities panel and connect it to the Start node.

3) Select the Write line activity. Now, type the following text into the Text box: “What is Your Name?”

4) Next, add a Delay activity and connect it to the Write line activity.

5) Select the Delay activity and go to the Properties panel.

6) In the Duration field, set 00:00:50. This is a 50-second delay between the two logged messages.

7) Take another Write line activity and connect it to the Delay activity.

8) In the Text field, write “My name is Ben.”

9) After clicking on the Run button, the Output panel shows the message that delays it by 50 seconds.

h) The Break activity:

Ans) The Break activity is used to break/stop the loop at a particular point, and then continue to the next activity according to the requirement.

Example:

1. Add a Sequence activity to the Designer panel.

2. Next, add a For each activity inside the Sequence.

3. Create two variables; an integer variable named item, and an array integer variable named x. Then, set them to the text field.

4. Now, assign a default value to the integer variable x.

5. Add a Break activity inside the body of the loop.

6. Under the For Each activity, add a Write line activity.

7. In the Write line activity, type item.ToString in the text field.

6) Illustrate the Step-by-step example using Sequence and Control flow to read an array of names and to count the number of names that start with the letter ‘a’.

Ans)

Perform the following steps:

1) Drag and drop a **Flowchart** activity from the Activities panel.

2) Drag and drop a **Sequence** activity inside the Flowchart and set it as **Start** node.

3) Double click on the Sequence activity.

4) Create a variable. Give it a name (in our case, we will create an array of type string and name the variable as names).

5) Set the variable type to **Array of [T]**. When asked for the type of array, select **String**.

6) Also, initialize the array in the Default section of the variable by giving it a default values.

For example, {“John”,”Sam”,”Andrew”,”Anita”}.

7) Create a variable of type integer **Count** for storing the result. Set the variable type to **Int32**.

8) Drag and drop a For each activity inside the Sequence.

9) Specify the array name in the expression box of the For each activity.

10) The For each activity will pick up one name from the array each time until it reaches the end.

11) Drag and drop the If activity from the Activities panel and place it inside the For each activity at the location where Drop activity here is mentioned.

12) Specify the condition in the expression box of the If activity.

13) The If activity is used to check for a particular condition/expression. If that expression is satisfied, then the block will be executed. Otherwise, the Else block will be executed.

14) We have specified the expression as **item.ToString.StartsWith(‘a’)**.

15) This expression specifies the name present in the item variable starts with the letter ‘a’.

16) The For each activity iterates over the array, picks up one name at a time, and stores it as a variable, item.

17) Now, we are going to use the Count variable and increment it each time a name from an array starts with the letter a.

18) For this, we have to use the **A+B Assign** activity.

19) Drag and drop the A+B Assign activity inside the If activity. Set the **To** property to **Count** and the **Value** property to **Count+1** of the A+B Assign activity.

20) Just drag and drop a Message box activity inside the Sequence activity.

21) Specify the count variable in the expression box of the Message box activity and convert it to a string by applying ‘.toString’ method.

22) Hit the Run button or press F5 and see the result.

7) Create a bot which does the following procedure:

a) Open Notepad and write some data into it.

Ans)

1) Drag and drop a Flowchart activity from the Activities panel.

2) Click on the Recording icon on the top of UiPath Studio.

3) A drop-down menu will appear with the options - Basic, Desktop, Web and Citrix, indicating the different types of recording. Select Desktop and click on Record.

3) Click on Notepad to open it. A Notepad window will pop up.

4) Click on the text area of Notepad. Type into the dialog box and check the empty field.

5) Press Enter. Data will be written on the Notepad text area.

b) Copy the data to the clipboard and extra the data from the clipboard.

Ans)

6) Click on the Edit button. A pop-up window will appear asking you whether you want to use an anchor.

7) As you can see clearly, the anchor element of the Edit button can be the File or Format button. In this case, we have chosen the Format button.

8) Then, it will automatically start recognizing the Edit button. Choose the Select all option from the drop-down list.

9) Once again, click on the Edit button. It will again ask you to indicate the anchor element.

10) Indicate the anchor button and the Edit button will be highlighted, giving you a drop-down box.

11) Select the Copy option. This copied text is now stored in the clipboard.

12) Double-click on the Recording sequence that is generated by the recording.

13) Scroll down and drag and drop the Copy selected text and Message box activities inside the Recording sequence.

14) Create a variable of type String to store the output value of Copy selected text.

15) This variable will receive the required text from the clipboard with the Copy selected text activity.

16) Now, specify the newly created variable in the Output property of the Copy selected text activity.

17) Specify the string variable in the text property of the Message box activity.

18) Hit the Run button to see the result.

8) Define data table. Illustrate the steps to build a data table and display its contents.

Ans) A data table is a tabular form of data structure. It contains rows and each row has columns.

First, create an empty project. Give it a proper name:

1) Drag and drop a Flowchart activity on the Designer panel.

2) Drag and drop a Sequence activity and set it as the Start node.

3) Drag and drop the Build Data Table activity inside the Sequence activity.

4) Click on the Data Table button. A pop-up window will appear on the screen. Remove both the columns by clicking on the Remove Column icon.

5) Add three columns by simply clicking on the + symbol.

6) Specify the column names and select the appropriate data types from the drop-down list. Click on the OK button.

7) We will add column **Name** of **String** Data Type, **Roll_No** of **Int32** type and finally **Class** of **String** type.

8) Now enter some values just to insert the data into the rows. Click on the OK button and our data table is ready.

9) We have to iterate over the data table's rows to make sure everything works correctly.

10) In order to store the Data Table created by Build Data Table activity, we have to create a data table variable data of DataTable type.

11) Also, assign the Output property of the Build Data Table activity with this variable.

12) Specify the data table variable's name there.

13) After our data table is ready, we will iterate the data table's rows to make sure everything works correctly.

14) Drag and drop the For each row activity from the Activities panel inside the Sequence activity.

15) Specify the data table variable's name (data) in the expression text box of the For each row activity.

16) Drag and drop a Message box activity inside the For each row activity.

17) In the Message box activity, Inside the message box we have to write following string:

row(“Name”).ToString+”-“+ row(“Roll_No”).ToString+”-“+ row(“Class”).ToString.

18) The row is the variable which holding data for the data row in each iteration.

19) This row variable contains all the columns of a particular row. Instead of the column name, we can also specify the column index.

20) Hit the Run button to see the result.

9) Define data scraping. Write the steps to build a bot to extract the data from amazon website.

Ans) Data scraping is a method in which we can dynamically create tabular data records of search items on the web.

Using data scraping, we can build the data table at runtime.

Let us consider an example of extracting data from Amazon's website.

Perform the following steps:

1) Drag and drop the Flowchart activity from the Activities panel.

2) Drag and drop the Sequence activity inside the Flowchart activity. Double-click on the Sequence activity.

3) Drag and drop the Open Browser activity inside the Sequence activity.

4) Specify the URL of amazon site in the text box.

5) Click on the Data Scraping icon on the top left corner of UiPath Studio.

6) A window will pop up. Click on the Next button.

7) Now, there will be a pointer pointing to the UI elements of the web page. Click on the name of the book.

8) It will ask you to point to a second similar element on the web page. Point to a second similar element on that web page.

9) Specify the name that you want to give for that extracted data column. Click on the Next button.

10) A list of names will appear in a separate window.

11) If you want to extract more information, then click on the Extract correlated data button and repeat the same process once again. Otherwise, click on the Finish Button.

12) It will ask you to locate the next page's button/link.

13) If you want to extract more information about the product and it spans across multiple pages, then click on the Yes button and point to the next page's button/link. Then, click on it.

14) If you want to extract only the current page's data, click on the No button, (you can also specify the number of rows that you want to extract data from: By default it is 100).

15) Data scraping generates a data table. (In this case, ExtractedDataTable is generated.)

16) Change the scope of ExtractedDataTable to the Flowchart so that it is accessible within the Flowchart activity.

17) Drag and drop the Output data table activity on the Flowchart.

18) Set the Output property of the Output data table activity as: ExtractedDataTable.

19) Connect the Output data table activity to the Data Scraping activity.

20) Drag and drop the Message box activity on the Designer window.

21) Also create a string variable to receive the text from the Output data table activity (in our case, we have created a result variable).

22) Specify the text property of the Output data table activity as the result variable to receive the text from the Output data table.

23) Connect the Message box activity to the Output data table activity.

24) Double-click on the Message box and specify the text property as the result variable.

25) Hit the Run button and see the result.

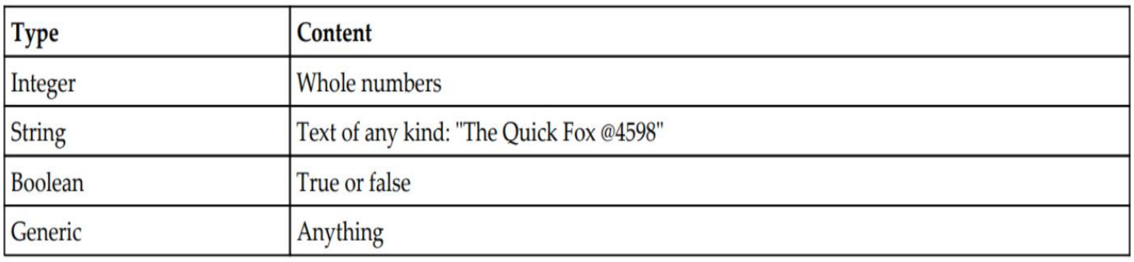
10) Explain the variable and scope of variable. State the different types of variables.

Ans) A variable is the name that is given to a block of memory and is used to hold data.

The Scope is the region under which the data has its effect or availability.

- In UiPath, we can declare a variable in the Variables section. Just give it a meaningful name and select the appropriate type from the drop-down list.
- A variable is used to store the data.
- We can choose the Scope of the variable according to your requirements.
- For example, if we wish to create a variable to store the name of a person, then we should declare Name: Andy.
- It is a good practice to create meaningful variable names. This becomes very useful in debugging the program.

The different types of variables are:



Type	Content
Integer	Whole numbers
String	Text of any kind: "The Quick Fox @4598"
Boolean	True or false
Generic	Anything

11) Explain how collections work.

Ans) In a collection, we can store one or more data points, but all the data must be the same.

Consider the following example:

1) Drag and drop a Flowchart activity onto the main Designer panel.

2) Drag and drop a Sequence activity inside the Flowchart and set it as Start node.

3) Create a variable in the Variables panel and give it a meaningful name (in this example, we have created a variable named arr, which is an array of integers) and choose the data type as an array of integers.

4) Initialize the array as {1,2,3,4,5} in the Default section. You can initialize it with the Int32 data type.

5) Drag and drop a For each activity from the Activities panel inside the Sequence, and drag and drop a Message box activity inside the For each activity.

6) Specify the array name in the expression text box of the For each activity.

7) Inside the Message box activity, specify the item variable and convert it to string using the ToString method.

8) Hit the Run button to see the result.

9) All the values will pop up at once.

12) Illustrate the process of creating a data table and then writing all its data to an Excel file.

Ans)

1) Drag and drop a Build data table activity from the Activities panel.

2) Double-click on this activity.

3) A window will pop up. Two columns have been generated automatically. Delete these two columns.

4) Add your column by clicking on the + icon and specify the column name. You can also select your preferred data type. You are free to add any number of columns.

5) In this project, we are adding two columns. The procedure for adding the second column is almost the same. You just have to specify a name and its preferred data type.

6) We have added one more column (Roll) and set the data type to Int32 in the data table.

7) We have also initialized this data table by giving some values to its rows.

8) Create a variable of type Data Table. Give it a meaningful name.

9) Specify this data table's name in the Data Table property of the Build data table activity.

10) Drag and drop the Excel application scope inside the main Designer window.

11) Specify the Excel sheet's path.

12) Connect this activity to the Build Data table activity.

13) Inside the Excel application scope activity, drag and drop the Write Range activity.

14) Specify the data table variable name that we created earlier and set it as a Data table property inside the Write Range activity.

15) We can also specify the range. In this case, we have assigned it as an empty string.

16) That's it. Hit the Run button or press F5 to see the result.