**Practical No: 1**

**Sequence and Flowchart**

**AIM: A) Create a simple sequence- based project.**

**Steps with output**

1. Create a New Sequence:
2. Add an Input Dialog Activity:  
   A screenshot of a computer

   Description automatically generated
3. Add a Message Box Activity:  
   A screenshot of a computer

   Description automatically generated

**OUTPUT:**

A screenshot of a computer

Description automatically generatedA screenshot of a computer screen

Description automatically generated

**AIM: B) Create a flowchart-based project.**

1. Create a New Flowchart:

A screenshot of a computer

Description automatically generated

1. Add the Input Dialog:

A screenshot of a computer

Description automatically generated

1. Add a Message Box:  
   A screenshot of a computer program

   Description automatically generated

**OUTPUT:**

A screenshot of a computer

Description automatically generatedA screenshot of a message box

Description automatically generated

**Learnings:**

**A. Create a Simple Sequence-based Project:**

Creating a new sequence, adding an Input Dialog, and a Message Box activity taught us the basics of building a linear workflow where user input is obtained and displayed in a MessageBox.

**B. Create a Flowchart-based Project:**

Building a new flowchart, incorporating an Input Dialog, and a Message Box activity helped us understand the flowchart structure, allowing for a more visual and branched project layout.

**Practical No: 2**

**Calculator | Types of Variables**

**AIM: A) Automate UiPath Number Calculation (Subtraction, Multiplication, Division of numbers).**

1. **Add Input Dialog for First Number:**
   1. Drag and drop the "Input Dialog" activity into your sequence.
   2. Configure the input dialog to prompt the user for the first number.

A screenshot of a computer

Description automatically generated

1. **Add Input Dialog for Second Number:**
   1. Add another "Input Dialog" activity into your sequence.
   2. Configure this input dialog to prompt the user for the second number.

A screenshot of a computer

Description automatically generated

1. **Add a "Message Box" activity to your sequence.  
   A screenshot of a computer

   Description automatically generated**
2. **Use expressions in the message box to display the results of arithmetic operations such as subtraction, multiplication, and division based on the user-provided numbers.  
   A screenshot of a computer

   Description automatically generated**
3. **Save your workflow and run the sequence.**

OUTPUT:

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Learnings:**

Using Input Dialogs, we took two inputs from the user for the first and second numbers. Subsequently, we performed addition, subtraction, multiplication, and division on those inputs and displayed the results individually in a Message Box. Additionally, we discovered how to incorporate a new line in a Message Box using Environment.NewLine.

**AIM: b) Create an automation UiPath project using different types of variables (number, datetime, Boolean, generic, array, data table)**

1. **Build Data Table Activity:**
   1. Use the "Build Data Table" activity to create a DataTable.

A screenshot of a computer

Description automatically generated

* 1. Add columns and set their data types (e.g., "RollNo" as Int32, "Name" as String).

A screenshot of a computer

Description automatically generated

A white rectangular object with a black border

Description automatically generated with medium confidence

1. **Output Data Table Activity:**
   1. Use the "Output Data Table" activity to convert the DataTable to a string.  
      A screenshot of a computer

      Description automatically generated
   2. Set the DataTableVar as the DataTable and create a new variable (e.g., TableVar) for the output.  
      A white rectangular object with a black border

      Description automatically generated with medium confidence
2. From the Variables tab, create variables for Number (numVar), DateTime (dateTimeVar), Boolean (boolVar), Generic (genericVar), and Array (stringArrayVar).  
   A screenshot of a computer

   Description automatically generated
3. Use multiple "Assign" activities to assign values to the variables.  
   A screenshot of a computer

   Description automatically generated
4. Use the "Message Box" activity to print all variables.  
   A screenshot of a computer

   Description automatically generated
5. Use another "Message Box" activity to print datatable.  
   A screenshot of a computer

   Description automatically generated

OUTPUT:

A screenshot of a computer

Description automatically generated A screenshot of a computer

Description automatically generated

**Learnings:**

Understanding the usage of the "Build Data Table" activity to create and configure a DataTable, defining columns with specific data types.

Practical application of different variable types (Number, DateTime, Boolean, Generic, Array) and DataTable, utilizing "Assign" activities, and displaying their values using the "Message Box" activity in UiPath.

**Practical No: 3**

**Decision Making and Looping statements.**

**AIM: A) Create an automation UiPath Project using Decision Making** **statements.**

**Steps with output**

1. **If Then**
2. Drag and drop an "Input Dialog" activity into the sequence.A screenshot of a computer

   Description automatically generated
3. Drag and drop an "If" activity below the "Input Dialog" activity.
4. Add a "Message Box" Activity (Then Branch):

A screenshot of a computer

Description automatically generated

1. Add a "Message Box" Activity (Else Branch)A screenshot of a computer

   Description automatically generated

Output:

A screenshot of a computer

Description automatically generatedA screenshot of a computer message

Description automatically generated

1. **Else IF**
2. Drag and drop an "Input Dialog" activity into the sequence**.**A screenshot of a computer

   Description automatically generated
3. Add a "Message Box"  
   A screenshot of a computer

   Description automatically generated A screenshot of a computer

   Description automatically generatedA screenshot of a computer

   Description automatically generated

**OUTPUT:**

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

1. **Switch**
2. Use an Input Dialog to get the user's choice.

**A screenshot of a computer

Description automatically generated**

1. Set up a Switch Activity. Input the variable for expression. Add cases with associated message boxes for each choice. Don't forget to include a Default case for any choices not covered.

**A screenshot of a computer

Description automatically generated**

**OUTPUT:**

**A screenshot of a computer

Description automatically generatedA screenshot of a computer screen

Description automatically generated**

**Learnings**

**A. If Then:**

By using an Input Dialog followed by an If activity, we learned to create a decision-making statement. If a certain condition (e.g., number > 0) is true, a MessageBox in the "Then" branch is executed; otherwise, the "Else" branch with another MessageBox is triggered.

**B. Else If:**

Employing an Input Dialog and a Message Box, we grasped the concept of an "Else If" statement. Depending on the condition, a specific message is displayed in the MessageBox, demonstrating an alternative path in the execution.

**C. Switch:**

Through an Input Dialog capturing user choice and a Switch activity, we learned to handle multiple cases efficiently. The Switch statement directs the flow based on the user's choice, with each case associated with a specific action in a MessageBox.

**AIM: B) Create an automation UiPath Project using looping statements**

**Steps with output**

1. **While Loop**
2. Add while activity and set condition to -> count <=5

A screenshot of a phone

Description automatically generated

1. Create variable count and set variable type to int32  
   A white rectangular object with black lines

   Description automatically generated
2. Inside while body create message box and write message -> "Current Count: " + count.ToString()  
   A screenshot of a computer

   Description automatically generated
3. Drag and drop assign activity inside while   
   Set the "To" field to Count.

Set the "Value" field to Count + 1.

A screenshot of a computer

Description automatically generated

**OUTPUT**

A screenshot of a computer message

Description automatically generatedA screenshot of a computer message

Description automatically generatedA screenshot of a computer message

Description automatically generatedA screenshot of a computer message

Description automatically generated

A screenshot of a computer message

Description automatically generatedA screenshot of a computer message

Description automatically generated

1. **Do While**
2. Do While Loop: Drag and drop a "Do While" activity from the "Activities" panel into your workflow.

A screenshot of a phone

Description automatically generated

1. Create a variable named ‘Count’ of type "Int32" to keep track of the current count.

A white screen with black text

Description automatically generated

1. we will use the condition: Count <= 5. This means the loop will continue as long as the Count variable is less than or equal to 5.
2. Inside the "Do While" activity, drag and drop a "Message Box" activity.

A screenshot of a computer

Description automatically generated

1. Configure the Message Box Activity:  
   A screenshot of a computer

   Description automatically generated
2. Add an "Assign" activity inside the "Do While" loop:

Set the "To" field to Count.

Set the "Value" field to Count + 1.

A screenshot of a computer

Description automatically generated

**OUTPUT**

**A screenshot of a computer message

Description automatically generated** **A screenshot of a computer message

Description automatically generated** A screenshot of a computer message

Description automatically generated A screenshot of a computer message

Description automatically generated**A screenshot of a computer error message

Description automatically generated** **A screenshot of a computer message

Description automatically generated**

1. **FOR EACH**

we'll create a "For Each" loop to iterate through a list of names and display each name using a message box.

1. Add a List of Names:

A screenshot of a computer

Description automatically generated

1. In the "Default" value field of the variable, enter the list of colors enclosed in curly braces {} and separated by commas.

A screenshot of a chat

Description automatically generated

1. Drag and drop a "For Each" activity from the "Activities" panel into your workflow.

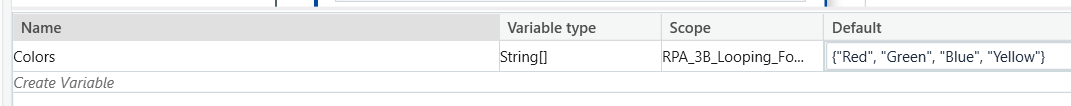
A screenshot of a computer

Description automatically generated

1. Inside the "For Each" activity, drag and drop a "Message Box" activity.

A screenshot of a computer

Description automatically generated



OUTPUT:

A screenshot of a computer message

Description automatically generated A screenshot of a computer

Description automatically generated A screenshot of a computer

Description automatically generated A screenshot of a computer

Description automatically generated

**Learnings**

**A. While Loop:**

The While Loop iterates as long as the count is less than or equal to 5, displaying the current count in a Message Box and incrementing the count in each iteration.

**B. Do While Loop:**

The Do While Loop continues executing as long as the count is less than or equal to 5, showing the current count in a Message Box and incrementing the count within the loop.

**C. For Each Loop:**

The For Each Loop iterates through a list of names, displaying each name in a Message Box, showcasing the functionality of iterating through collections using a loop.

**Practical No: 4**

**Desktop & Web Recording**

**AIM: A) Automate any process using desktop recording.**

**Steps with output**

1. Click on recording -> Desktop recording
2. Start app -> choose word app
3. Click on record

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generatedA screenshot of a computer

Description automatically generated

**OUTPUT**

**A screenshot of a computer

Description automatically generated**

**AIM: B) Automate any process using web recording.**

**Steps with output**

**Step 1:** Add the Open Browser activity to the sequence and input the URL [www.imdb.com]. Additionally, maximize or snap the window.

A screenshot of a computer

Description automatically generated

**Step 2:** Utilize the Click activity to select the search bar. Follow it up with the Type Into activity to input the movie name.A screenshot of a computer

Description automatically generated

**Step 3:**Repeat the Click activity to select the movie panel. Proceed to use the Get Text activity to fetch the rating. [Generate a string variable and input it into the Get Text properties.]

A screenshot of a computer

Description automatically generated

**Step 4:** Show the rating by utilizing a Message Box.A screenshot of a computer program

Description automatically generated

**OUTPUT:**

A screenshot of a computer message

Description automatically generated

**Learnings:**

**A. Automate any process using Desktop Recording:**

By initiating desktop recording and automating steps like starting the Word app, we learned the basics of capturing and automating desktop processes.

**B. Automate any process using Web Recording:**

Through web recording, we learned how to automate web-based processes, such as opening a browser, interacting with elements like search bars and movie panels, extracting information using the Get Text activity, and displaying the result in a Message Box.

**Practical No: 5**

**ARRAY**

**AIM: A) Consider an array of names. We have to find out how many of them start with the letter "a". Create an automation where the number of names starting with "a" is counted and the result is displayed.**

**Steps with output**

1. Create the Counter Variable:
   1. Name the variable as CountA.
   2. Set the variable type to "Int32"

A white rectangular object with a black border

Description automatically generated

1. Create a Name Variable:
   1. Name the variable (e.g., Names).
   2. Set the variable type to "Array of [your chosen data type]" -> "Array of String."

A screenshot of a computer error

Description automatically generated

1. Add an Assign Activity: (enter any names with {} )A screenshot of a computer

   Description automatically generated
2. Drag and drop a "For Each" activity into the sequence.A screenshot of a cell phone

   Description automatically generated
3. Add an If Activity (Inside For Each):

currentItem.ToLower().StartsWith("a")

A screenshot of a computer

Description automatically generated

1. Inside the "Then" section of the "If" activity, add an "Assign" activity.A screenshot of a computer

   Description automatically generated
2. Add a Message Box Activity (After For Each):A screenshot of a computer

   Description automatically generatedA screenshot of a computer

   Description automatically generated

A screenshot of a computer

Description automatically generated

**OUTPUT:**

A screenshot of a message box

Description automatically generated

**Learnings:**

**A. Count Names Starting with "a":**

We learned to count the number of names starting with "a" in an array by creating and using a counter variable, employing a For Each loop, and implementing an If activity to check and increment the count accordingly. The result was then displayed in a Message Box.

**Practical No: 6**

**Excel Automation**

**AIM: A) Create an application automating the read, write and append operation on excel file**

**ENABLE CLASSIC OPTION**

**READ**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**OUTPUT**

A screenshot of a computer

Description automatically generated A screenshot of a computer

Description automatically generated

**WRITE**

1. Open main workflow
2. Activities -> Excel application scope
3. Add path of excel file  
   A screenshot of a computer

   Description automatically generated
4. In Do -> Add build data table -> add table data -> create variable(myData)
5. Add write range  
   A screenshot of a computer

   Description automatically generated

**OUTPUT**

A screenshot of a computer

Description automatically generated

**Append**

1. Open main workflow
2. Activities -> Excel application scope
3. Add path of excel file  
   **A screenshot of a computer

   Description automatically generated**
4. In Do -> Add Read Range -> Sheet2 -> create variable(myData)
5. Add Append range  
   **A screenshot of a computer

   Description automatically generated**

**OUTPUT**

A table with numbers and letters

Description automatically generated

**AIM: B) Automate the process to extract data from an excel file into a data table and vice versa.**

1. **Excel to Datatable**
2. Prepare an Excel file with data.  
   A screenshot of a computer

   Description automatically generated
3. Use the "Excel Application Scope" activity to specify the Excel file location.  
   A screenshot of a white box

   Description automatically generated
4. Read Range:
   1. Add the "Read Range" activity inside the Excel Application Scope.  
      A screenshot of a computer

      Description automatically generated
   2. Create a variable (e.g., DataTableVar) to store the output DataTable.  
      A white rectangular object with a black border

      Description automatically generated
5. Output DataTable:
   1. Use the "Output Data Table" activity.
   2. Set the DataTableVar as the DataTable.  
      A screenshot of a computer

      Description automatically generated
   3. Create a new variable (e.g., TableVar) for the output.  
      A white rectangular object with a black border

      Description automatically generated
6. Add a "Message Box" activity.  
   A screenshot of a computer

   Description automatically generated

**OUTPUT:**

**A screenshot of a computer

Description automatically generated**

1. **Datatable to excel**
2. Use the "Excel Application Scope" activity to specify the Excel file location.  
   A screenshot of a phone

   Description automatically generated
3. Add the "Build DataTable" activity.  
   A screenshot of a computer

   Description automatically generated
4. Define the structure of the DataTable (columns).  
   A screenshot of a computer

   Description automatically generated
5. Create a variable (e.g., Data2) to store the DataTable.  
   A white rectangular object with a black border

   Description automatically generated
6. Add the "Write Range" activity inside the Excel Application Scope.  
   A screenshot of a computer

   Description automatically generated
7. Configure it to write the DataTable (Data2) to a specific sheet.  
   A white rectangular object with black lines

   Description automatically generated with medium confidence
8. Save your workflow and run the sequence.

**OUTPUT:**

A screenshot of a computer

Description automatically generated

**LEARNING:**

Automating Excel data extraction involves specifying file locations, reading and outputting DataTables, and handling DataTable structures, enhancing efficiency in data manipulation.

**Practical No: 7**

**AIM: A) Implement the attach window activity.**

1. Use the "Open Application" activity to launch Notepad.  
   A screenshot of a computer

   Description automatically generated
2. Add the "Attach Window" activity to identify and attach to the Notepad window.  
   A screenshot of a computer

   Description automatically generated
3. Within the "Do" section of the "Attach Window" activity, include the "Type Into" activity.
4. In the "Type Into" activity, input some text within quotation marks to be typed into the Notepad.  
   A screenshot of a computer

   Description automatically generated

**OUTPUT  
A screenshot of a computer

Description automatically generated**

**Learnings:**

**A. Implement Attach Window Activity:**

By using the Attach Window activity, we learned to launch an application (Notepad), attach to its window, and perform actions within that window, such as typing text using the Type Into activity.

**AIM: B) Find different controls using UiPath.**

**Steps with output**

1. Utilize the "Open Browser" activity and input the LinkedIn URL.  
   A screenshot of a phone

   Description automatically generated
2. Add an "Anchor Base" activity to locate elements relative to a reference anchor.  
   A white rectangular object with a blue stripe

   Description automatically generated
3. Apply the "Find Element" activity as an anchor, indicating the "Email or phone" text on the login page.  
   A close-up of a box

   Description automatically generated
4. Integrate the "Type Into" activity within the "Anchor Base," targeting the email text box indicated by the anchor. Input the email within quotation marks.  
   A screenshot of a computer

   Description automatically generated

A screenshot of a computer

Description automatically generated **OUTPUT:  
A screenshot of a computer

Description automatically generated**

**Learnings:**

**B. Finding Different Controls:**

Using Open Browser, Anchor Base, Find Element, and Type Into activities, we learned how to locate and interact with specific elements on a web page, in this case, typing into the email text box on the LinkedIn login page.

**AIM: C) Demonstrate the following activities in UiPath:**

1. **Mouse (click, double click and hover)**
2. Create an existing Word file with content.  
    **A black and grey rectangular object with a white x

   Description automatically generated with medium confidence**
3. Use the Double Click activity to open the Word file from File Explorer. **A screenshot of a computer

   Description automatically generated**
4. Apply the Send Hotkey activity (Ctrl + A) to select all text in the Word document. **A screenshot of a computer

   Description automatically generated**
5. Utilize the Click activity to click on the "Highlight" option in the Word application. **A screenshot of a computer

   Description automatically generated**
6. Use the Hover activity to hover over the "Heading 1" option in the ribbon of the Word application. **A screenshot of a computer

   Description automatically generated**

**OUTPUT**

**A yellow rectangle with black text

Description automatically generatedA screenshot of a computer

Description automatically generated**

1. **Type Into**
2. Add type into activity after performing above steps
3. Type some text into with quotation marks.  
     
   A screenshot of a computer

   Description automatically generated

**OUTPUT:**

**A yellow box with black text

Description automatically generated**

1. **Type Secure Text**
2. Generate a fresh variable and modify its type to SecureString.  
   **A screenshot of a computer

   Description automatically generated**
3. Populate the variable with a Username/Password format using the VB Expression: new system.net.NetworkCredential("", "Password123").SecurePassword  
   **A screenshot of a computer

   Description automatically generated**
4. Integrate a Type Secure Text activity, targeting a password field within a web browser. Ensure to assign the previously created SecureString to the relevant property.  
   **A screenshot of a computer

   Description automatically generatedA white and grey striped background

   Description automatically generated with medium confidence**

**OUTPUT:**

**A screenshot of a computer

Description automatically generated**

**Learnings:**

**C(i). Mouse Activities:**

We explored Mouse activities in UiPath, including Double Click to open a Word file, Send Hotkey to select all text, Click to highlight text, and Hover to interact with the ribbon.

**C(ii). Type Into Activity:**

We practiced using the Type Into activity to input text, enhancing our understanding of text entry automation.

**C(iii). Type Secure Text:**

We learned to secure sensitive information by creating a SecureString variable and using the Type Secure Text activity to input a password securely into a designated field in a web browser.

**Practical No: 8**

**Triggering events, Screen Scraping, Plug-ins**

**Aim: A) Demonstrate the following events in UiPath:**

**i. Element triggering event**

**ii. Image triggering event**

**iii. System Triggering Event.**

1. **Element Triggering event**
2. Utilize the "Click Trigger" activity to set up a mouse click event.
3. Configure the mouse button to be the left mouse button.  
   A screenshot of a computer

   Description automatically generated
4. Set the window scope for the click trigger by selecting the Notepad window.  
   A screenshot of a computer

   Description automatically generated
5. Inside the event, include the "Type Into" activity to specify the text you want to be written upon the mouse click.  
   A screenshot of a computer

   Description automatically generated

**OUTPUT:**

A screenshot of a computer

Description automatically generated

1. **Image Triggering event**
2. Drag and drop the "Image Trigger" activity into your sequence.  
   A screenshot of a computer

   Description automatically generated
3. Click "Indicate Scope on screen" and choose a part of an image by dragging.
4. Add a follow-up activity, like "Message Box," which will pop up when you click on the selected part of the image.  
   A screenshot of a computer

   Description automatically generated

**OUTPUT:**

A screen shot of a message

Description automatically generated

1. **System Triggering Events**
2. Include the "System Trigger" activity in your workflow.  
   A screenshot of a computer

   Description automatically generated
3. Integrate a follow-up activity, such as "Message Box," to occur after the System Trigger. This activity will trigger when any key or mouse press occurs.  
   A screenshot of a computer

   Description automatically generated

OUTPUT:

A screenshot of a message box

Description automatically generated

Learnings:

**i. Element Triggering Event:**

We learned to set up a mouse click event using the "Click Trigger" activity, configuring the mouse button, defining the window scope, and incorporating a follow-up "Type Into" activity to execute upon the mouse click.

**ii. Image Triggering Event:**

We explored the "Image Trigger" activity to create an event triggered by clicking on a selected part of an image, and added a follow-up "Message Box" activity to display a message when the image is clicked.

**iii. System Triggering Events:**

By using the "System Trigger" activity, we learned to set up an event triggered by any key or mouse press and added a follow-up "Message Box" activity to demonstrate the occurrence of the triggering event.

**Aim: b) Automate the following screen scraping methods using UiPath**

**i. Full Text**

**ii. Native**

**iii. OCR**

1. **Full Text**
2. Drag and drop the "Get Full Text" activity into your sequence.
3. Click "Indicate on Screen" within the activity.
4. Select a text pane on the screen.
5. In the properties of the "Get Full Text" activity, add a string variable to the output using [Ctrl + K].
6. Add a "Message Box" activity to your sequence.
7. Use the variable created in step 4 to print the extracted text in the message box.  
   A screenshot of a computer

   Description automatically generated

OUTPUT:

A screenshot of a message box

Description automatically generated

1. **OCR**
2. Drag and drop the "Get OCR Text" activity into your sequence.
3. Click "Indicate on Screen" within the activity.  
   A screenshot of a computer

   Description automatically generated
4. Select an image containing text that you want to extract.
5. Create a variable to store the OCR output.
6. Add a "Message Box" activity to your sequence.  
   A screenshot of a computer

   Description automatically generated
7. Use the variable created in step 3 to display the OCR-extracted text.

OUTPUT:

A screenshot of a computer program

Description automatically generated

**Learnings:**

**i. Full Text:**

We learned to use the "Get Full Text" activity to extract text from a specific pane on the screen, store it in a variable, and display it using a Message Box.

**ii. OCR:**

By employing the "Get OCR Text" activity, we learned to capture text from an image, save it in a variable, and showcase the extracted text in a Message Box, demonstrating the application of Optical Character Recognition (OCR) in UiPath.

**Aim: 3. Install and automate any process using UiPath with the following plug-ins:**

1. **PDF Plugin**
2. Install UiPath.PDF Package:
   1. Open UiPath Studio.
   2. Navigate to "Manage Packages" > "All Packages."
   3. Search for "uipath.pdf" and select 'UiPath.PDF.Activities.'
   4. Click "Install" to install the package.

A screenshot of a computer

Description automatically generated

1. Drag and drop the "Read PDF Text" activity into your sequence.
2. Within the activity, click on the field for selecting a PDF file and choose the desired PDF.  
   A screenshot of a computer

   Description automatically generated
3. In the properties of the "Read PDF Text" activity, set the range of pages and select a string variable to store the extracted text.
4. Integrate a "Message Box" activity into your sequence.
5. Use the string variable to display the extracted text in the message box.  
   A screenshot of a computer

   Description automatically generated

OUTPUT:

A screenshot of a computer

Description automatically generated

1. **EXCEL PLUGIN**

**Pre-requisite: Install UiPath.Excel.Activities**

1. Install Excel Plugin(Home > Tools > UiPath Extensions > Install ‘Excel Add-in')
2. Create an Excel file containing two columns: "Name" and "Marks."
3. In UiPath Studio, add an "Excel Application Scope" activity to your sequence.
4. Within the activity, select the Excel file you created in step 2.  
   A screenshot of a computer

   Description automatically generated
5. Add a "Read Range" activity inside the Excel Application Scope.
6. Integrate a "Message Box" activity into your sequence.  
   A screenshot of a computer

   Description automatically generated
7. Use the output of the "Read Range" activity to display the read data in the message box.

OUTPUT:

A screenshot of a computer

Description automatically generated

**Learnings:**

**i. PDF Plugin:**

We learned to install the UiPath.PDF package, use the "Read PDF Text" activity to extract text from a PDF file, and display the extracted text in a Message Box.

**ii. Excel Plugin:**

We grasped the process of installing the UiPath.Excel.Activities plugin, adding an Excel file containing data, utilizing "Excel Application Scope" and "Read Range" activities to read data, and displaying the read data in a Message Box using the output variable.

**Practical No: 9**

**Email Automation**

**AIM: A) Automate the process of send mail event (on any email)**

1. Drag and drop the "Use Gmail" activity into your sequence.  
   A screenshot of a computer

   Description automatically generated
2. Choose the default option and authenticate your Gmail account in the browser pop-up.  
   A screenshot of a computer

   Description automatically generated
3. Inside the "Do" section of the "Use Gmail" activity, add a "Send Email" activity.
4. Select Gmail as the account.
5. Enter the recipient’s email address, subject, body, and attach a file if necessary.  
   A screenshot of a computer

   Description automatically generated

**OUTPUT:**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a email

Description automatically generated** **A screenshot of a computer

Description automatically generated** A screenshot of a computer

Description automatically generated A screenshot of a computer

Description automatically generated

**Learnings:**

**A. Automate Send Mail Event:**

We learned to use the "Use Gmail" activity to send an email, authenticate a Gmail account, and configure email details, enabling automation of the email-sending process.

**B:** **Automate the process of launching an assistant bot on a keyboard event**

1. Drag and drop the "Trigger Scope" activity into your sequence.
2. Inside the Trigger Scope, add a "Hotkey Trigger" activity.
3. Configure the Hotkey Trigger by selecting the desired hotkey combination.  
   A screenshot of a computer

   Description automatically generated
4. In the Action section of the Trigger Scope, add the sequence of activities that you want to be executed when the specified hotkey combination is pressed.  
   A screenshot of a computer

   Description automatically generated
5. Save your workflow and run the sequence.

**OUTPUT:**

**A screenshot of a computer

Description automatically generated**

**B. Automate Launching Assistant Bot on Keyboard Event:**

We grasped the concept of using the "Trigger Scope" and "Hotkey Trigger" activities to execute a sequence of actions when a specified hotkey combination is pressed, facilitating automation triggered by keyboard events.

**Aim: c) Demonstrate the Exception handing in UiPath.**

1. Drag and drop the "Try Catch" activity into your sequence.
2. Inside the Try block, add an activity that might throw an exception during execution. For example, use a "Type Into" activity to type text into an already running Notepad window. If the Notepad window is not running, it will throw an exception.  
   A screenshot of a computer

   Description automatically generated
3. In the Catch block, handle a specific exception. For example, take a "SelectorNotFoundException" exception.
4. Add a "Message Box" activity inside the Catch block to notify you when an exception is thrown.  
   A screenshot of a computer

   Description automatically generated
5. In the Finally block, add a "Message Box" activity to test if the Try-Catch activity successfully handles the exception.  
   A screenshot of a computer

   Description automatically generated
6. Save your workflow and run the sequence.

**OUTPUT:**

**Case 1:** **Notepad is running**

**A screenshot of a computer screen

Description automatically generated**

**Case 2: Notepad is not in an active state**

(In this situation, the UiPath application will pause for a duration of 30 seconds, attempting to find the Notepad window. If unsuccessful, it will then trigger an exception.)

**A screenshot of a computer message

Description automatically generatedA screenshot of a message box

Description automatically generated**

**Learnings:**

**C. Demonstrate Exception Handling:**

We understood the implementation of the "Try Catch" activity for handling exceptions, demonstrated by attempting to type into a Notepad window and catching a specific exception (SelectorNotFoundException), with notification via a "Message Box" activity. The Finally block tested the overall success of handling exceptions.