TERMWORK 1

THEORY:

Printing "Hello World" using a Message Box is a simple introductory exercise that helped us to understand the basic workflow design and activity usage within the UiPath Studio. The exercise involves displaying a pop-up message with the text "Hello World" using the Message Box activity.

Activities Used:

Message Box Activity: The Message Box activity is a UiPath core activity that displays a pop-up message to the user. It is commonly used for displaying information, notifications, or debugging purposes during the automation workflow development. In this case, the Message Box activity will be utilized to display the "Hello World" message.

Packages:

In this specific exercise, you will not require any additional packages as the Message Box activity is a built-in activity within UiPath Studio. However, it's essential to understand that UiPath supports a wide range of packages that extend the functionality and capabilities of the platform. These packages provide additional activities and features for automating various tasks and interacting with different software systems and applications.

TERMWORK 2

THEORY:

This termwork focuses on demonstrating how UiPath's activities can be used to facilitate user input, variable assignment, and displaying results through message boxes.

Input Dialog Activity: The Input Dialog activity is utilized to prompt the user for input during the automation process. It displays a dialog box where the user can enter values for the variables to be swapped. In this case, the input dialog is used to obtain the initial values of the variables from the user.

Assign Activity: The Assign activity is employed to assign values to variables within the workflow. It allows users to manipulate variables and perform operations on them. In the context of the variable swap termwork, the Assign activity is used to assign the input values to the respective variables and carry out the swapping operation using a temporary variable.

Message Box Activity: The Message Box activity is used to display information or output during the automation process. In the variable swap termwork, the Message Box activity is utilized to showcase the swapped values of the variables to the user.

TERMWORK 3

THEORY:

The workflow aims to demonstrate different input methods in automating data input in an application, specifically Microsoft Word instead of Notepad due to limitations encountered with Windows 11 Notepad automation.
The input methods employed in this workflow are Type Into, Simulate Click, and SendWindowMessages. By utilizing these methods, the workflow showcases how data can be entered, modified, and formatted within the Word application.

Activities Used:

Word Application Scope: The Word Application Scope activity is used to establish a connection with the Microsoft Word application. It allows the subsequent activities to interact with Word.

Type Into Activity: The Type Into activity is used to input text into the Word document. In this workflow, it is utilized to type the phrase "automation makes life easier" into the document.

Simulate Click Activity: The Simulate Click activity is employed to simulate a mouse click on a specific element within the Word application. Here, it is used to maximize the Word window, ensuring the entire document is visible.

SendWindowMessages Activity: The SendWindowMessages activity is used to send keyboard input directly to the Word application window. It is used to input the phrase "welcome to the world of automation" into the document.

Get Text Activity: The Get Text activity is used to extract the current font size of the selected text in the Word document. It captures the font size for further manipulation.

Assign Activity: The Assign activity is used to store the font size retrieved from the Get Text activity into a variable. It allows for further calculations and modifications.

Type Into Activity (for updating font size): The Type Into activity is employed to update the font size by adding 5 to the retrieved font size value. It is used to change the font size of the selected text.

Send Hotkey Activity: The Send Hotkey activity is used to send a keyboard shortcut, specifically the "Enter" key. It is utilized to apply the updated font size to the selected text.

Packages:

The workflow may not require additional packages beyond the core UiPath activities. However, to interact with Microsoft Word effectively, I used the following package:

UiPath.Word.Activities: This package provides a set of activities specifically designed for automating tasks within the Microsoft Word application. It offers activities for document manipulation, formatting, content extraction, and more.

TERMWORK 4

THEORY:

This termwork involves building a workflow that automates the process of filling out a form on the RPAChallenge.com website using organized data from an Excel file. The workflow utilizes UiPath activities to access the Excel file, read the data, interact with the website, and populate the form fields dynamically.

Activities Used:

Excel Application Scope: The Excel Application Scope activity establishes a connection with the Excel application and enables the subsequent activities to access and manipulate the data in the Excel file.

Read Range Activity: The Read Range activity is used to read the data from the Excel file into a DataTable. It retrieves the organized data required for populating the form on the RPAChallenge.com website.

For Each Row Activity: The For Each Row activity enables the iteration over each row of the DataTable, allowing the workflow to process the data for filling out the form.

Click Activity: The Click activity is used to initiate the form-filling process on the RPAChallenge.com website. It interacts with the "Start" button on the website to begin filling the form.

Anchor Base and Find Element Activities: Within each Anchor Base activity, the Find Element activity is used to set an anchor element, which provides a stable reference point for locating the corresponding text box. The Find Element activity ensures that the workflow accurately identifies the target text box on the webpage.

Type Into Activity: Inside the Anchor Base activities, the Type Into activity is used to input the data from the Excel sheet into the respective text boxes on the website. The Type Into activity populates the form fields dynamically with the values from the Excel file.

Click Activity (Submit Button): After the Parallel activity, a Click activity is used to interact with the "Submit" button on the website, completing the form-filling process.

Packages:

Following packages are used, depending on the specific needs and activities utilized:

UiPath.Excel.Activities: This package provides activities for interacting with Excel files, such as reading data from worksheets, writing data to worksheets, and performing various Excel-related operations.

UiPath.UIAutomation.Activities: This package offers activities for automating interactions with the user interface of applications and web browsers. It includes activities for clicking, typing, finding elements, and more.

TERMWORK 5

(5a) THEORY:

The workflow is designed to interact with users and gather information about their eye color. It utilizes the Switch activity in UiPath Studio to handle different eye color choices provided by the user. Based on the input, the workflow displays the corresponding personality trait associated with the selected eye color using message boxes.

Activities Used:

Flowchart: The workflow is structured using a flowchart activity, which provides a visual representation of the logical flow of activities.

Input Dialog Activity: The Input Dialog activity is used to prompt the user to input their eye color choice. The input is expected as an integer value (1, 2, 3, or 4) corresponding to the available options.

Switch Activity: The Switch activity is used to evaluate the user's input and execute different branches of the workflow based on the selected eye color.

Message Box Activities: The Message Box activity is utilized within each case of the Switch activity to display the personality trait associated with the selected eye color. The message boxes provide a visual output to the user.

Packages:

The workflow does not require any additional packages beyond the default activities provided by UiPath Studio. The activities used, such as Flowchart, Input Dialog, Switch, and Message Box, are part of the core activities available in UiPath.

TERMWORK 5

(5b) THEORY:

The following termwork involves building a workflow using a While loop to determine if a user-inputted number is a prime number or not. The workflow utilizes a flowchart in UiPath Studio and incorporates activities such as Input Dialog, While, Assign, If, and Message Box. It provides a user-friendly way to check the primality of a given number and provides appropriate feedback based on the result.

Activities Used:

Flowchart: The flowchart activity serves as the structure for organizing the workflow, allowing for visual representation and easy navigation between different activities.

Input Dialog Activity: The Input Dialog activity prompts the user to enter a number. The user's input is stored in the "ipVal" variable, which will be evaluated for primality.

While Activity: The While activity implements a loop that continues until a specific condition is met. In this case, the loop continues as long as the value of "i" is less than or equal to "ipVal".

Assign Activity: The Assign activity is used to assign values to variables within the workflow. In this workflow, the Assign activity is used to increment the "flag" variable when the condition "(ipVal Mod i) = 0" is met, indicating a factor of the number.

If Activity: The If activity evaluates the condition "flag = 2" to determine if the number is prime or not. If the condition is true, it means that the number has only two factors (1 and itself) and is therefore prime. If the condition is false, it means that the number has more than two factors and is not prime.

Message Box Activity: The Message Box activity is used to display messages to the user. In this workflow, it is used to show whether the inputted number is a prime number or not.

Packages:

The workflow does not require any additional packages beyond the default activities provided by UiPath Studio. The activities used, such as Flowchart, Input Dialog, While, Assign, If, and Message Box, are part of the core activities available in UiPath.

TERMWORK 6

THEORY:

The following termwork involves building a workflow that utilizes various string manipulation methods to extract key information from a given text and display it in a different format. The workflow employs activities such as Assign and Message Box and incorporates methods like Format, Join, IndexOf, Split, and Substring. This automation aims to showcase the power of UiPath Studio in processing and transforming text data efficiently.

Activities Used:

Assign Activity: The Assign activity is used to assign values to variables within the workflow. In this termwork, the Assign activity is utilized to store and manipulate the given text, extract specific information, and format it for display.

Message Box Activity: The Message Box activity is employed to display the formatted information extracted from the text. It provides a user-friendly interface to present the results to the user.

Methods Used:

Format Method: The Format method is used to format the extracted information and create a formatted string for display. It allows for the insertion of variables and values into a specified pattern or template.

Join Method: The Join method is used to combine multiple string elements into a single string. It is utilized to concatenate specific words or phrases extracted from the text.

IndexOf Method: The IndexOf method is employed to determine the position of a specific character or substring within a string. It is utilized to identify the starting point for extracting relevant information from the text.

Split Method: The Split method is used to split a string into an array of substrings based on a specified delimiter. It enables the extraction of individual words or phrases from the given text.

Substring Method: The Substring method is employed to extract a specific portion of a string based on the starting and ending indices. It allows for the retrieval of desired segments of the text.

Packages:

The workflow does not require any additional packages beyond the default activities provided by UiPath Studio.

TERMWORK 7

THEORY:

The following termwork involves building a workflow that utilizes data table activities in UiPath Studio to join two library databases based on matching student IDs. The workflow reads data from two Excel sheets, performs an inner join operation using the Join Data Table activity, and displays the joined output in a message box. This automation aims to showcase the capabilities of UiPath Studio in handling and manipulating data tables to perform database-related tasks efficiently.

Activities Used:

Excel Application Scope: The Excel Application Scope activity is used to specify the Excel file ("data.xlsx") and perform operations on it, such as reading data from specific sheets and writing data to other sheets.

Read Range: The Read Range activity is utilized to read the data from the specified sheets ("Sheet1" and "Sheet2") of the Excel file. It retrieves the data and stores it in data tables (table1 and table2) for further processing.

Join Data Table: The Join Data Table activity is employed to join the two data tables (table1 and table2) based on the matching student IDs. By specifying the join conditions and type, it combines the data from both tables into a single output data table ("outputData").

Write Range: The Write Range activity is used to write the joined output data table ("outputData") to a specified sheet ("Sheet4") in the Excel file. This activity ensures that the joined data is saved for future reference or analysis.

Output Data Table: The Output Data Table activity is utilized to store the joined output data table ("outputData") in a variable ("output") for further usage or display.

Message Box: The Message Box activity is employed to display the joined output data table ("output") in a pop-up message box. It provides a user-friendly interface to present the results of the data joining operation.

Packages:

The workflow utilizes the default activities provided by UiPath Studio and does not require any additional packages for data table operations.

TERMWORK 8

(8a) THEORY:

This termwork involves building a workflow in UiPath Studio that utilizes the Screen Scraper Wizard to scrape text from a web page using the Full-Text scraping method. The scraped text is then stored in a Notepad file. This automation showcases the capabilities of UiPath Studio in extracting and saving specific text content from web pages.

Activities Used:

Open Browser: The Open Browser activity is used to navigate to the desired web page (www.google.com) for text scraping. It provides the starting point for interacting with the web content.

Type Into: The Type Into activity is used to enter the search term "UiPath" into the search box of the web page. It simulates the typing action to perform a search.

Screen Scraping Wizard: The Screen Scraping Wizard is a powerful tool in UiPath Studio for extracting data from various applications and web pages. In this workflow, it is used with the Attach Browser activity to target the specific browser window. The Get Full Text option is selected to scrape the desired text content from the web page.

Write Text File: The Write Text File activity is used to write the scraped text content into a Notepad file. It specifies the output file path and content to be written.

Packages:

The workflow utilizes the following package:

UiPath.UIAutomation.Activities: This package provides activities related to UI automation and screen scraping. The Screen Scraping Wizard is part of this package, allowing the extraction of text content from web pages.

TERMWORK 8

(8b) THEORY:

This termwork involves building a workflow in UiPath Studio that utilizes the Data Scraping Wizard to scrape blog post titles from the UiPath Blog across multiple pages. The automation will navigate to the UiPath Blog website, extract the structured data using the Data Scraping Wizard, and save the scraped information into an Excel file. This workflow showcases the power of UiPath Studio in automating the extraction of specific data elements from websites.

Activities Used:

Open Browser: The Open Browser activity is used to launch the web browser and navigate to the UiPath Blog website (www.uipath.com/blog). It serves as the starting point for interacting with the web content.

Message Box: The Message Box activity is used to display a notification message to indicate the start of the scraping process. It provides a visual confirmation to the user that the scraping will begin.

Data Scraping Wizard: The Data Scraping Wizard is a powerful tool in UiPath Studio that enables the extraction of structured data from web pages. In this workflow, it is used with the Attach Browser activity to target the specific browser window. The wizard assists in selecting the relevant data elements (blog post titles) and configuring the scraping parameters.

Excel Application Scope: The Excel Application Scope activity is used to open an Excel file ("blogData.xlsx"). It establishes the connection to the Excel application and allows subsequent activities to interact with the spreadsheet.

Write Range: The Write Range activity is used within the Excel Application Scope to write the scraped data into the Excel file. It specifies the target worksheet and the data table variable ("ExtractDataTable") that contains the scraped blog post titles.

Packages:

The workflow utilizes the following package:

UiPath.Excel.Activities: This package provides activities related to Excel automation, allowing the interaction with Excel files, worksheets, and data. The Excel Application Scope and Write Range activities are part of this package.

TERMWORK 9

THEORY:

The following termwork involves building a workflow in UiPath Studio that utilizes the Read PDF Text activity to extract email IDs and phone numbers from a PDF file. The extracted data is then stored in an MS Word file. This automation showcases the capabilities of UiPath Studio in extracting specific information from PDF documents and manipulating MS Word files.

Activities Used:

Read PDF Text: The Read PDF Text activity is used to read the text content from the specified PDF file ("challenge.pdf"). It extracts the text data and stores it in the variable "extractedData" for further processing.

Message Box: The Message Box activity is employed to display the extracted text data ("extractedData") in a pop-up message box. This activity provides a way to verify the correctness of the extracted data during the workflow execution.

Word Application Scope: The Word Application Scope activity is utilized to create a Word document and perform operations on it. It specifies the MS Word file ("extractedData") to work with.

Append Text: The Append Text activity is used to add the extracted data ("extractedData") to the MS Word document. This activity ensures that the extracted email IDs and phone numbers are appended to the existing content of the Word file.

Packages:

UiPath.PDF.Activities: This package provides the Read PDF Text activity, which is essential for extracting text data from the PDF file.

UiPath.Word.Activities: This package includes activities for working with MS Word files, such as Word Application Scope and Append Text. It allows the automation to interact with and manipulate Word documents.

TERMWORK 10

THEORY:

This termwork involves building a workflow that extracts attachments from emails with the subject containing the word "Resume". The workflow utilizes activities such as input dialog, get password, get IMAP mail message, for each, if condition, message box, and save attachments. The goal is to retrieve specific emails and save their attachments to a designated folder.

Activities Used:

Input Dialog: Prompts the user to enter their email username.

Get Password: Retrieves the password associated with the email.

Get IMAP Mail Message: Connects to the email server using IMAP protocol, retrieves emails from the "inbox" folder, and retrieves the top 10 emails.

For Each: Iterates over each email in the collection.

If Condition: Checks if the subject of the email contains the keyword "Resume".

Message Box: Displays a message stating that the keyword was found in the subject of the email.

Save Attachments: Saves the attachments from the email to a designated folder.

Packages Used:

UiPath.Mail.Activities: Required for the "Get IMAP Mail Message" and "Save Attachments" activities.