Term Work 1: Variable Swapping **Date:**

Objective:

Build a workflow that swaps two numbers using a third variable.

- Ask the user to input two numeric values and store them in two variables.
- Swap values of both the variables using a third variable.
- Display initial and swapped values of both the variables in the Output panel.

Process Overview:

- START
- Use an Input Dialog activity to receive two numeric values from the user.
- Store the received values in two integer variables called **First_Input_Value**, and **Second_Input_Value**
- Declare a third integer variable called **Swapping_Support_Variable**
- Use Assign activity to assign the value of **First_Input_Value** to **Swapping_Support_Variable**
- Use second Assign activity to assign the value of **First_Input_Value** to **Second_Input_Value**
- Use third Assign activity to assign the value of **Second_Input_Value** to **Swapping_Support_Variable**
- Use a Write Line activity to display initial and final values of **First_Input_Value** and **Second_Input_Value** in the Output panel.
- STOP

Term Work 2: Notepad application **Date:**

Objective:

Build a workflow that uses different Input Methods to input data in a Notepad.

- Open a Notepad file and type “Automation makes life easier”.
- Minimize the Notepad file using the ‘Click’ activity.
- Type “Welcome to the new world of Automation”

Process Overview:

- START
- Use Open Application activity to indicate a Notepad file.
- Use Type Into activity to enter “Automation makes life easier”.
- Minimize the Notepad window using the Click activity
- Use Type Into activity to enter “Welcome to the new world of automation”.
- Use Keyboard shortcuts activity to send “Ctrl + A”.
- STOP

Term Work 3.1: If Statement **Date:**

Objective:

Build a workflow using an If statement, which asks a user, whether the user will get the second Marshmallow or not.

- Ask the user, “Do you want to eat your first Marshmallow now or after 5 minutes?”
- If the user answers “Now”, respond with “Oops! You will not get the second Marshmallow.”
- If the user answers “After 5 minutes”, respond with “Congrats! You will also get the second Marshmallow.”
- If the answer is other than “Now” or “After 5 minutes”, respond with “Invalid Input”.

Process Overview:

- START
- Use an Input Dialog activity to ask the user “Do you want to eat your first Marshmallow now or after 5 minutes?”
- Store user response in a string variable.
- Use an If activity to check the user response o If the answer is “Now”, use a Message Box activity to display “Oops! You will not get the second Marshmallow.”
    - o If the answer is “After 5 minutes”, use a Message Box activity to display “Congrats! You will also get the second Marshmallow.” o If the answer is other than “Now” or “After 5 minutes”, use a Message Box activity to display “Invalid Input”.
- STOP

Term Work 3.2: Switch Activity **Date:**

Objective:

Build a workflow using Switch activity that asks users' their eye color and display their personality in a message box.

- Ask the user for their eye color.
- If the user enters "Blue", respond with "You must be very Brave!"
- If the user enters "Green", respond with "You must be Generous!"
- If the user enters "Gray", respond with "You must be very Wise!"
- If the user enters "Black", respond with "You must be very Bold!"

Process Overview:

- START
- Use an Input Dialog activity to get the eye color input of the user.
- Use a Switch activity to compare the input with four different cases – Blue, Green, Gray, and Black.
- Use Message Box activities to display the result of each case o For "Blue", display "You must be Brave!" o For "Green", display "You must be Generous!" o For "Gray", display "You must be very Wise!" o For "Black", display "You must be very Bold!"
- STOP

Term Work 4.1: Do While Loop **Date:**

Objective:

Build a workflow for a 'Guessing Game' with the following conditions:

- Generate a random number and prompt the user to input a number.
- In case of a wrong input, a message is displayed to the user stating, 'Please enter a lesser/greater number'.
- The loop keeps on running until the input number equals to the generated number.

Process Overview:

- START
- Use an Input Dialog activity within a Do While activity to get the guessed number from the user.
- For Do While activity, set the condition to check guessed number is not equal to the actual number.
- Use a Message Box activity to display “You Guessed it correct” for the correct match.
- Use an If activity within the Do While loop to check if the guessed number is equal to the actual number.
    - If correct, use a Message Box activity to display “You Guessed it correct” for the correct match.
    - Use another If activity within the Else section to check if the guessed number is greater than the actual number.
        - If correct, use a Message Box activity to display “Please try a smaller number”.
        - If incorrect, use a Message Box activity to display “Please try a greater number”.
- STOP

Term Work 4.2: While Loop **Date:**

Objective:

Build a workflow using a While loop that tells the user if the input is a prime number or not.

- Ask the user to input a number.
- Check if it is a prime number.
- If the input number is prime, then display "It is a prime number" in a message box.
- If the input number is not prime, then display "It is not a prime number" in a message box.

Process Overview:

- START
- Use an Input Dialog activity and ask for any number from the user and store in a variable called intNumber.
- Create two more variables intRandom and c with Variable Type as Int32 and Default value as 2 and 0 respectively in the variables panel.
- Use a While activity and set the condition to intRandom<Number.
- Use an If activity within the While activity and set the condition to intNumber mod intRandom=0.
- Use an Assign activity within the Then section and increment value of intCount by 1.
- Use an Assign activity after/below the If activity, and increment value of intRandom by 1.
- Use another If activity after/below the While activity and enter condition intCount>0.
- Use a Message Box activity within the Then section to display “It is not a prime number”.
- Use a Message Box activity within the Else section to display “It is a prime number”.
- STOP

Term Work 4.3: For Each Loop **Date:**

Objective:

Build a workflow to display file names from a folder in the Output panel and also store names in an MS Word file.

- Locate and select a folder containing multiple files.
- List the directory path of all the files in the Output panel.
- Also, store the updated names in an MS Word file and save and close it.

Process Overview:

- START
- Use a Select Folder activity to select a folder containing a few files.
- Use an Assign activity to store file names in an array.
- Use an Attach Window activity below the Assign activity and select MS Word window.
- Use a For Each activity to iterate through each file name in the array.
- Use a Write Line activity within the For Each activity to display file names in the Output panel.
- Use a Type Into activity below the Write Line activity to store file names in an MS Word file.
- Use Click and Send Hotkey activities to save and close the file.
- STOP

Term Work 5: Data Manipulation **Date:**

Objective:

Build a workflow using Format, Join, IndexOf, Split, and Substring methods that extract key information from a text and prints in a different format.

- Use the text "You always wanted to study Automation Training. The materials are available in the following places: UiPath Blog, UiPath Academy." for extraction.
- Extract "Automation Training" from the first sentence.
- Extract "UiPath Blog" and "UiPath Academy" from the second sentence.
- Display "get Automation Training from: UiPath Blog; UiPath Academy" in a message box.

Process Overview:

- START
- Use an Assign activity for the initial value of the message string variable: "You always wanted to study Automation Training. The materials are available in the following places: UiPath Blog, UiPath Academy."
- Create a new String variable **study** and use a succession of String methods to assign the course from the query:
  message.Split("."c).First.ToString.Substring(message.LastIndexOf("study"))
  - **Split("."c).First.ToString** extracts the first sentence of the String and converts it to a String
  - **Substring(message.LastIndexOf("study"))** extracts the Substring starting from "get"
- Create a new List variable **places** and use a succession of String methods to assign the places from the query:
  message.Split("."c)(1).ToString.Split(":"c).Last.ToString.Split(","c).ToList
  - **message.Split("."c)(1).ToString** extracts the second sentence of the String and converts it to a String.
  - **Split(":"c).Last.ToString** splits the remaining string and keeps only the last

part of it.

- **Split(","c).ToList** takes each string separated by comma and adds it as an element in the List variable.

- Use a Message Box activity to display output using this expression:

String.Format("{0} from: {1}", study ,String.Join(";", places))

- **String.Join** is used to extract each element in the "places" List variable and display them.

- STOP

Term Work 6: Data Table Manipulation Date:

Objective:

Build a workflow using data table activities to join two library databases using matching student ID and display the output in a message box.

- Create a data table variable and populate it with student ID and name of students.
- Create another data table variable, and populate it with student ID and book names
- Join both the data tables based on matching student ID.
- Remove the student ID column and sort the final data table as per student names in alphabetical order from A to Z.
- Display the final data table containing the student and book names in a message box as a string.

Process Overview:

- START
- Use two Build Data Table activities to create two tables. Store them in two DataTable variables called **dt_users** and **dt_overdueBooks**.
    - **dt_users** variable contain ID of students and name of user as string.
    - **dt_overdueBooks** variable contain ID of students and name of books as string.
- Use a Join Data Table activity. Choose the Inner type for the Join activity. Write the two column names to be used as Join criterion and create a new data table variable to store the output called **dt_borrowedBooks**
- Use a Remove Data Column activity to delete duplicate column – student ID – by specifying its index.
- Use a Sort Data Table activity to sort the data table based on the name of students in alphabetical order from A to Z.
- Use an Output Data Table activity to print the content of the data table to a String variable.
- Use the Message Box activity to display the output.
- STOP

Term Work 7: Basic recording **Date:**

Objective:

Build a workflow to open a notepad, type some text and change font, size and style using basic recording.

Process Overview:

- START
- Open UiPath Studio and create a process. Name it as “basic recording”
- Add a Sequence activity and name it “Sequence – basic recording”.
- Open notepad in background.
- Add a Use application/browser activity in sequence and name it as “Open application- open notepad for recording”.
- Click on “Indicate application to automate” and click the image file opened. The path will be automatically saved in the Application path.
- Select the Do section of Use application/browser activity, click on App/Web Recorder button from design tab.
- Select the typing area on text file as the Target and click on confirm button.
- A text box is prompted to type the text that has to be typed in text file. Click on confirm after typing the text.
- After the text is typed into the file, click on Format button from menu and click on Confirm, then click on Font from the dropdown list and click on Confirm. Font window will be opened.
- Hover the mouse on Font Textbox and choose type into option. Type “Times New Roman. Similarly type “Italic” in Font style and 20 in size.
- Then click on Ok and confirm it.
- Now save the recording and return to studio.
- Reset the changes in notepad and run the workflow. See the output in text file.
- STOP

Term Work 8.1: Web recording **Date:**

Objective:

Build a workflow to automate shopping using web recording.

Process Overview:

- START
- Open UiPath Studio and create a process. Name it as “web recording”
- Add a Sequence activity and name it “Sequence – web recording on amazon”.
- Create two variables emailaddress and password as string type and set the default value with valid id and password.
- Open a browser and open a webpage using URL www.amazon.in.
- Add a Use application/browser activity in sequence and name it as “Open application- open www.amazon.in for recording”.
- Click on “Indicate application to automate” and click the webpage opened. The path will be automatically saved in the Application path.
- Select the Do section of Use application/browser activity, click on App/Web Recorder button from design tab.
- Click on Sign in button and click on confirm button.
- A text box is prompted to type the email address. Hover the mouse onto that text box and choose ‘type into’ option.
- Choose ‘Expression’ type for text and type the variable name “emailaddress” and click on confirm.
- Click on ‘Continue’ and confirm it.
- A text box is prompted to type the password. Hover the mouse onto that text box and choose ‘type into’ option.
- Choose ‘Expression’ type for text and type the variable name “password” and click on confirm.
- Click on ‘Sign in’ and confirm it.
- Now, hover the mouse on ‘Search amazon.in’ tab and choose ‘type into’ option.
- Type ‘decoupage paper+[k(Enter)]’ and confirm it.
- Click on the first item that is listed and click on ‘Buy now’ button.
- It will redirect to cart.
- Now save the recording and return to studio.
- Reset the changes in amazon.in website and run the workflow. See the output in browser.
- STOP

Term Work 8.2: Screen scraping **Date:**

Objective:

Build a workflow that extract text from a text image and writes that text into a text file.

Process Overview:

Initial settings:

1) Go to Project -> Settings-> OCR -> DefaultOCREngine.

Set the Run value and Debug value as Uipath Screen OCR.

2) Open browser and type the URL as cloud.uipath.com.
   On top left corner, click on Product Launcher-> Admin -> Licenses -> Robot & services. Copy the API key under computer vision.
   Go to Project -> Settings-> OCR -> Uipath Screen OCR ->ApiKey
   Paste the API key copied from the website into run value and debug value under ApiKey then click on Ok.

- START
- Open UiPath Studio
- Add a Sequence activity and name it "Sequence – Screen OCR"
- Open a text image file in background.
- Add a Use application/browser activity in sequence and name it as "Open application- Open image file to extract text".
- Click on "Indicate application to automate" and click the image file opened. The path will be automatically saved in the Application path.
- In the Do section of Use application/browser activity, add Get OCR Text activity and name it as "Get OCR text- indicate text in image file and extract it".
- Click on "Indicate Element on Screen" and click on the text area on the image.
- Go to Properties panel of Get OCR Text activity. Right click in the text box for Text under Output and click on create variable and name the variable as ocrtext.
- Open a notepad file in which the extracted text will be written.
- Add Set Text activity below Use application/browser activity and name it as "Set Text- Write extracted text to text file".
- In Text textbox in Set Text activity, type the variable name ocrtext.
- Now save the workflow and run it. See the output in text file.
- STOP

Term Work 9: Exception handling **Date:**

Objective:

Build a workflow using a Try Catch activity to do the following:

- Take Name, Gender, and Age as the user input.
- Subtract current year with Age value to get the Year of Birth.
- Handle an error that occurs due to a reckless user input of an incorrect age containing the 11-digit number.
- Continue the process to display the Name, Gender, and Year of Birth of the user in a message box.

Process Overview:

- START
- Use three Input Dialog activities within the Try Catch activity to ask for the Name, Gender, and Age of the user.
- Use an Assign activity to subtract the age from the current year to get the year of birth of the user
- Use Exception Type: System.Exception in the Catches section of the Try Catch activity to handle reckless input from the user. Store error in a string variable.
- Use a Message Box activity to display the Name, Gender, and Year of Birth of the user along with the Error, if any.
    - STOP

Open Ended Date:

Objective:

Build a workflow if, string split method and set text to text file that writes user input to notepad file based on few conditions.

Ask the user to input their email id.

- Check if the email id belongs to gmail.edu domain
- Read other details if yes and write details to a text file

Process Overview:

- START
- Use an Input Dialog activity to ask the email id.
- Split the email id using @ character and extract the domain name.
- Compare domain name with “git.edu” using if condition.
- If it’s true, then ask user to enter semester using input dialogue.
- Check if semester is equal to 8. If yes, then ask user to input name and CGPA.
- Use set text activity to write email id, name and CGPA to text file.
- If domain is false or semester is not 8, then display appropriate message.
- STOP

Term Work 10.1: **Date:**

Sending email using UiPath + Gmail

Objective:

Send a simple email from your Gmail account using UiPath with Gmail's secure app password.

Process Overview:

🔹 Step 1: Enable 2-Step Verification (One-Time Only)

1. Open: 👉 https://myaccount.google.com/security

2. Under "Signing in to Google", click 2-Step Verification

3. Set up using your mobile number

4. Once set, it will show "2-Step Verification: ON"

🔹 Step 2: Generate Gmail App Password

1. Go to 👉 https://myaccount.google.com/apppasswords

2. Under Select App → choose Mail

3. Under Select Device → choose Windows Computer (or "Other" and type UiPath)

4. Click Generate

5. Copy the 16-character password shown (remove any spaces)

🔹 Step 3: Create a New Project in UiPath

1. Open UiPath Studio

2. Click Process → Name it GmailSender → Create

🔹 Step 4: Add Required Package

1. Go to Manage Packages (top ribbon)

2. Under Official, search and install:

✅ UiPath.Mail.Activities

🔹 Step 5: Add and Configure Send SMTP Mail Message

1. In your Main.xaml → Add a Sequence

2. Drag in Send SMTP Mail Message activity

3. Configure the properties:

Property Value To : "receiver@gmail.com"

Subject : "Test Mail from UiPath"

Body: "Hi, this is a test mail."

Email: "your_email@gmail.com"

Password: "your_16_char_app_password" (in quotes)

Server: "smtp.gmail.com"

Port: 587

SecureConnection :Auto or StartTls

🔻 Step 6: Run

Click Run

Check your Sent Mail in Gmail — the email should appear!

Term Work 10.2: **Date:**

Objective:

Access 5 unread emails from Inbox of a Gmail account using UiPath.

Process Overview:

Step 1: Open UiPath and create a flowchart.

Step 2: Add Get IMAP Email List Activity and connect it to start symbol in flowchart.

Step 3: Go to properties panel of Get IMAP Email List activity and set the following properties:

1. Turn off “Use Integration Service”
2. Server: “imap.gmail.com”
3. Port: 993
4. Email: your_emailid@gmail.com
5. Password: "your_16_char_app_password" (in quotes)
6. Unread only: True
7. Limit email to First: 5
8. Right click in Output Email list textbox and click on create variable.
   Set Var: Mails

Step 4: Add For Each activity below Get IMAP Email List and connect to it.

Step 5: Go to properties panel of For Each activity. Type variable name Mails in textbox of In*Property. Change the item name to Mail.

Step 6: Double click on For Each activity. Add Message Box activity in Body section.

Step 7: In Text input type Mail.Subject.ToString.

Step 8. Save and Run the flowchart to see 5 unread emails on message box.