RPA Design and Development

v4.0

UiPath

Lesson 5 ¦ Control Flow

Control Flow – Exam Topics

1. Describe the functionality of the Control Flow activities (for example: If, Switch, While, Do While, For Each, etc.) and workflow types

2. Use Sequence and Flowchart layouts in projects.

3. Use the IF, Flow Decision, Else If activities in projects and the Vb.Net If Operator.

4. Use the For Each, While, Do While and Switch activities.

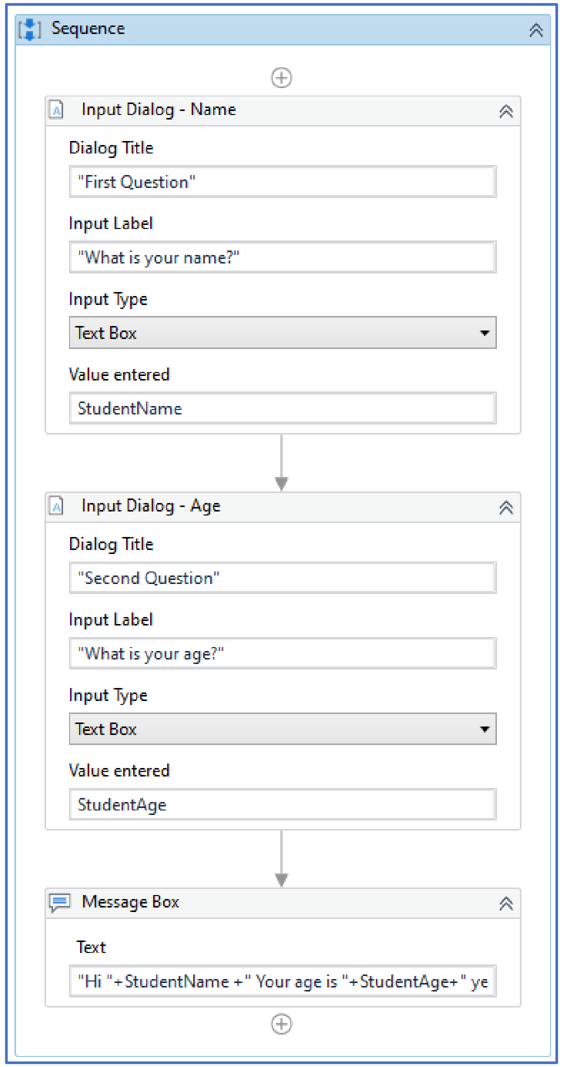
Sequences

A Sequence is a container in which activities are placed one after another and executed linearly.

A Sequence:

- Enables the user to create linear processes
- Enables seamless movement from one activity to another
- Can be reused

Example: A Sequence that asks for the Name and Age of a user and displays a greeting message



Sequence

Input Dialog - Name

Dialog Title
"First Question"

Input Label
"What is your name?"

Input Type
Text Box

Value entered
StudentName

Input Dialog - Age

Dialog Title
"Second Question"

Input Label
"What is your age?"

Input Type
Text Box

Value entered
StudentAge

Message Box

Text
"Hi "+StudentName +" Your age is "+StudentAge+" ye

Control Flow and Its Types

- Control Flow
- Types of Control Flow

Control Flow

Control Flow is the order in which activities or actions are executed in a workflow.

Control Flow Activities

- Used to define the decisions to be made during the execution of a workflow

- Allows users to define rules and execute conditional statements within the project

- Found in the Activities panel, under Workflow > Control

Types of Control Flow

The flow of a program can be controlled in two ways:

Decision-based

- A decision is made based on a specific condition
- If the condition is met, the program executes one part of the workflow, and if the condition is not met, then it executes the other part
- **Example: If** and **Switch** activities

Iteration-based

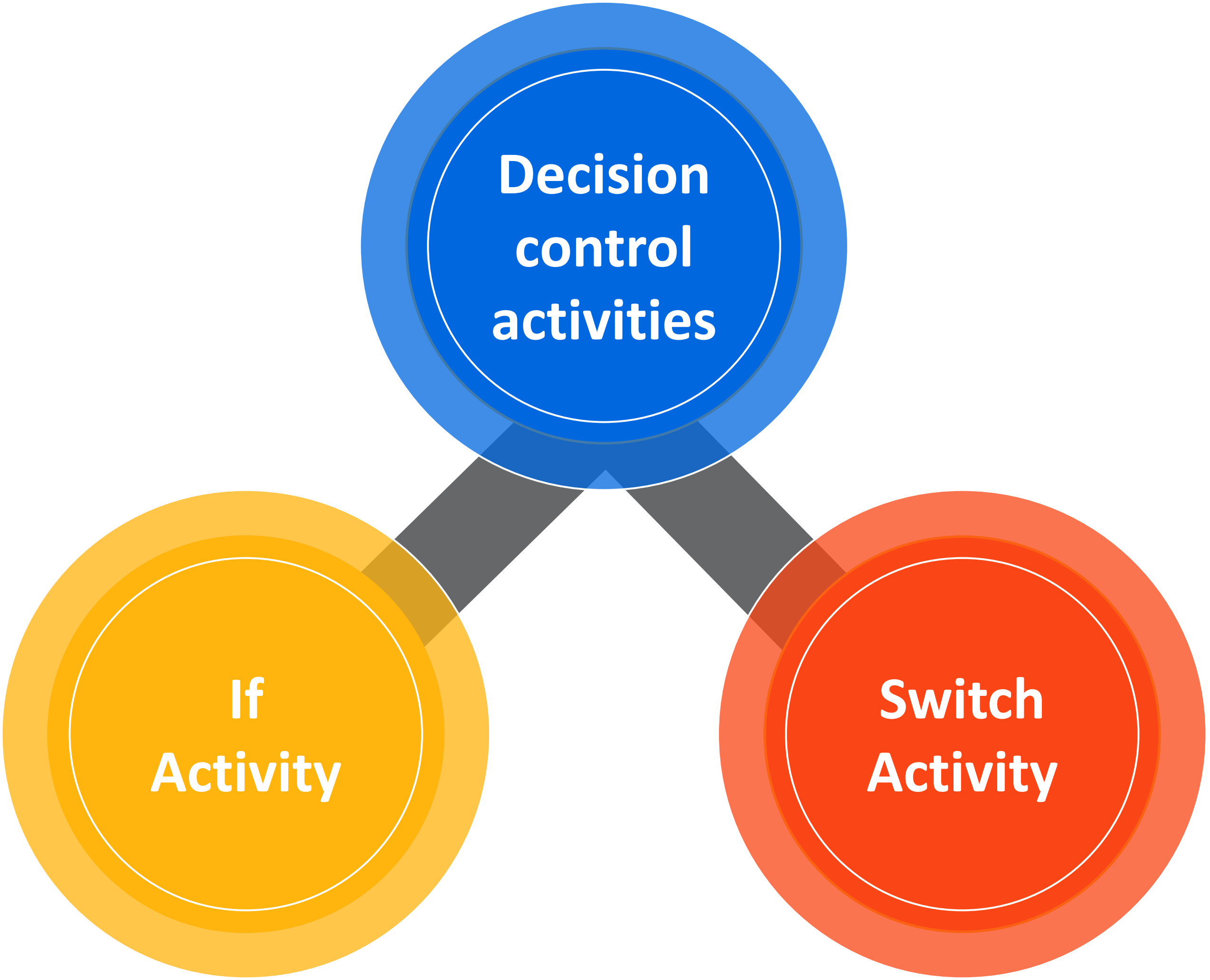
- A particular part of a program is executed multiple times until a specific condition is met or holds true
- **Example: While**, **Do While**, and **For Each** activities

Decision Control

- Introduction to Decision Control
- If Activity
- Switch Activity
- If Activity vs. Switch Activity

Introduction to Decision Control

Helps the user to decide the execution flow of a program based on certain conditions.



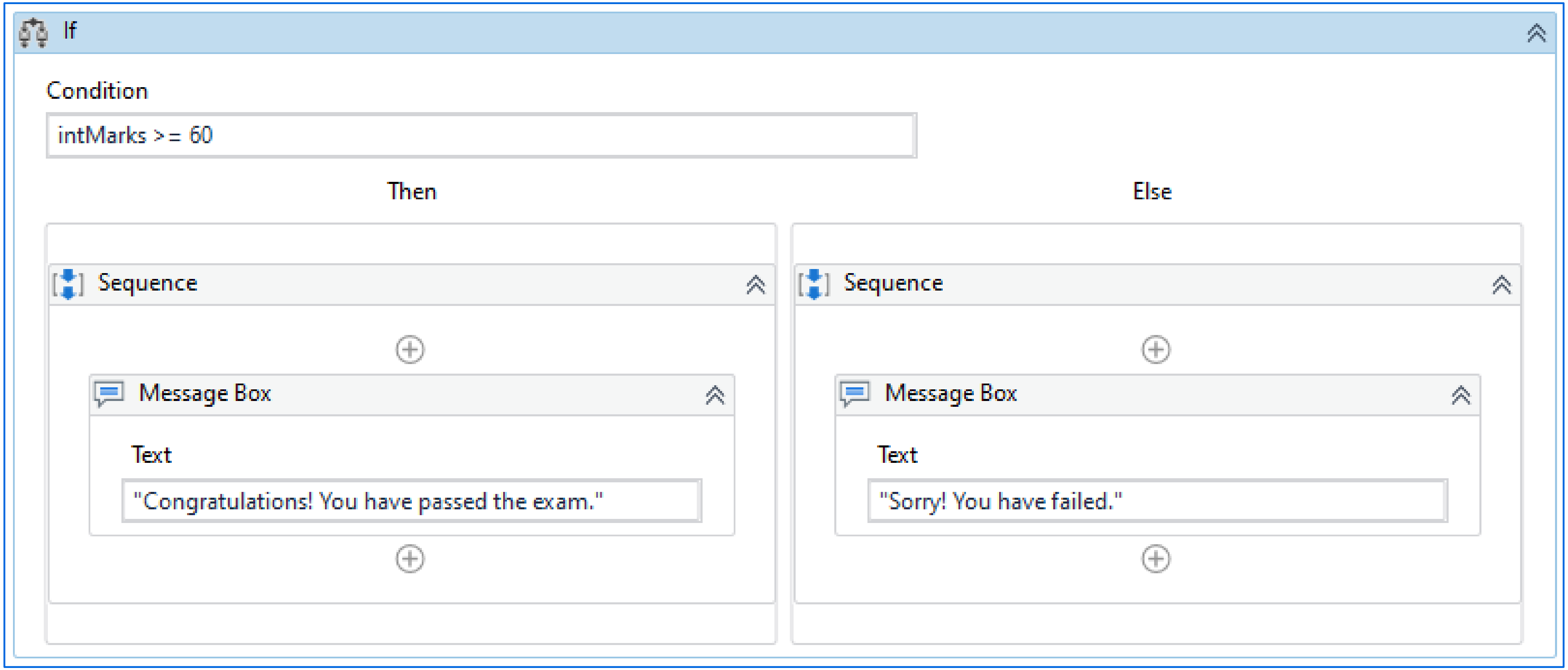
Decision control activities

If Activity

Switch Activity

If Activity

Contains a statement with a condition and two sets of instructions (Then & Else) as outcomes.



If

Condition

intMarks >= 60

Then

Sequence

Message Box

Text

"Congratulations! You have passed the exam."

Else

Sequence

Message Box

Text

"Sorry! You have failed."

UiPath™ Academy

Chapter 4:

Control Flow in Studio

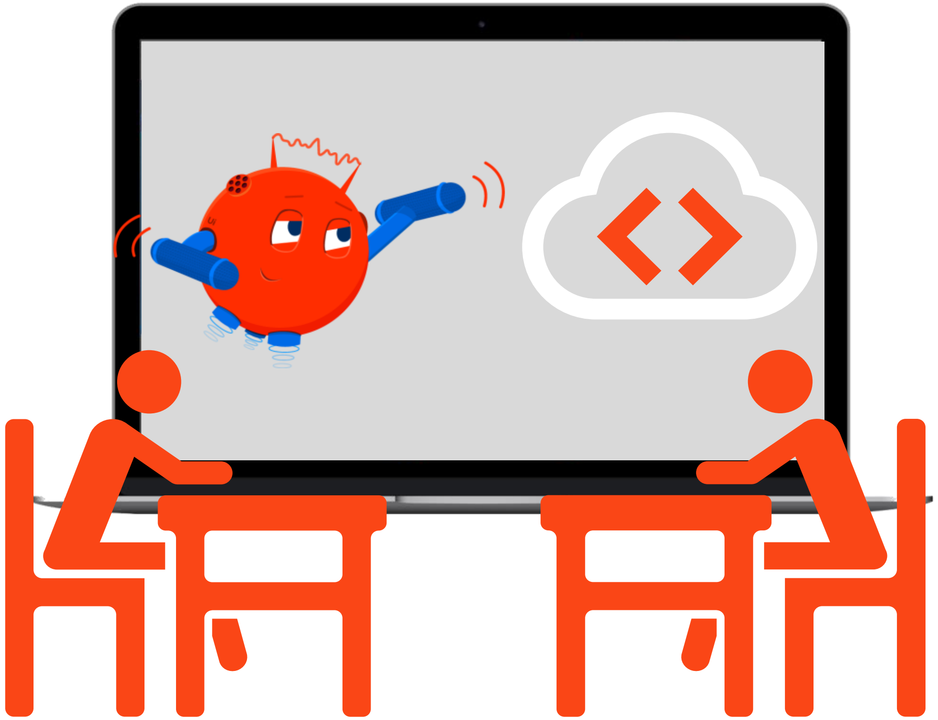
The If Statement

Classroom Exercise

Demonstrate the use of **If** activity by building a workflow that informs whether the user has passed the exam.

- Take user’s name and marks obtained in an exam as input
- The passing marks are greater than or equal to 60
- Display the result in a message box for Pass as "Congratulations! you have passed the exam."
- Display the result in a message box for Fail as "Hello User, you have failed the exam". Replace "User" with the user's name

Practice Exercise - If activity

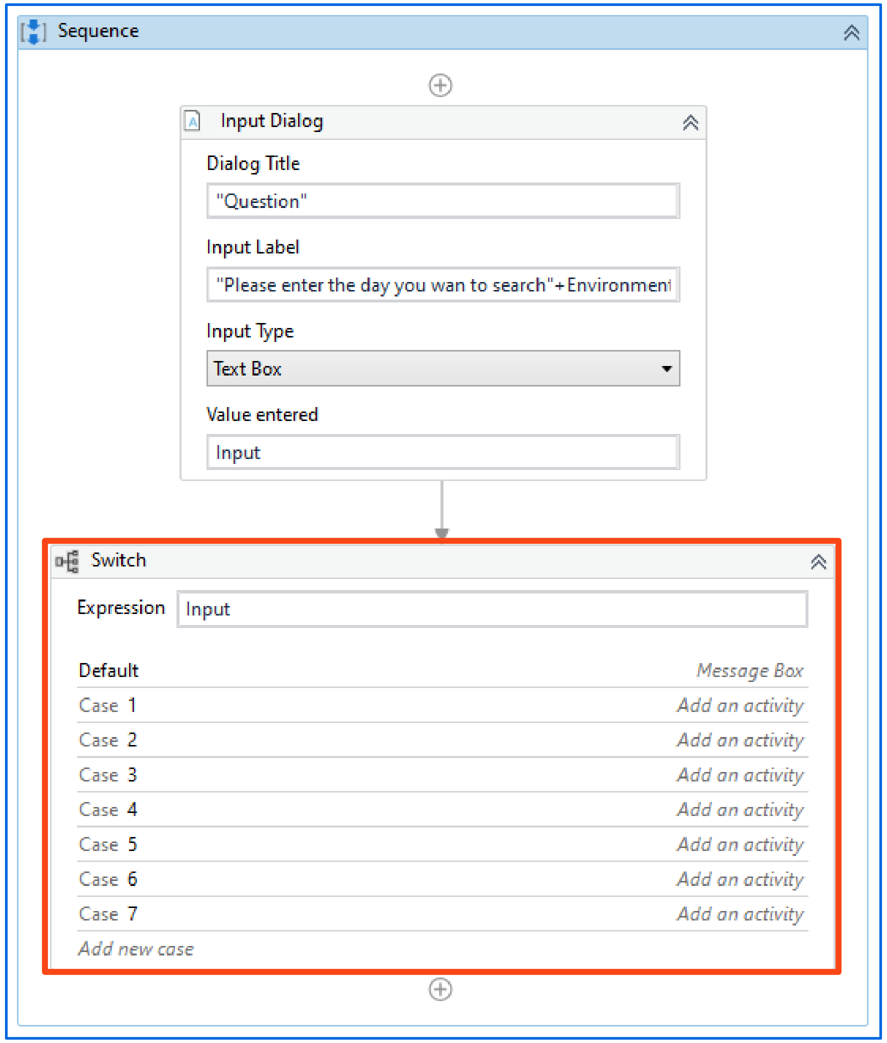


Build a workflow using an **If** activity, which asks the user, whether the user will get the second Marshmallow.

- Ask the user, “When do you want to eat your first Marshmallow? Choose from the following options: 1. Now, 2. 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”

Switch Activity

Executes one choice out of multiple statements based on the value of a specific expression.



Sequence

Input Dialog

Dialog Title
"Question"

Input Label
"Please enter the day you wan to search"+Environment

Input Type
Text Box

Value entered
Input

Switch

Expression: Input

Default — Message Box
Case 1 — Add an activity
Case 2 — Add an activity
Case 3 — Add an activity
Case 4 — Add an activity
Case 5 — Add an activity
Case 6 — Add an activity
Case 7 — Add an activity
Add new case

UiPath™ Academy

Chapter 4:

Control Flow in Studio

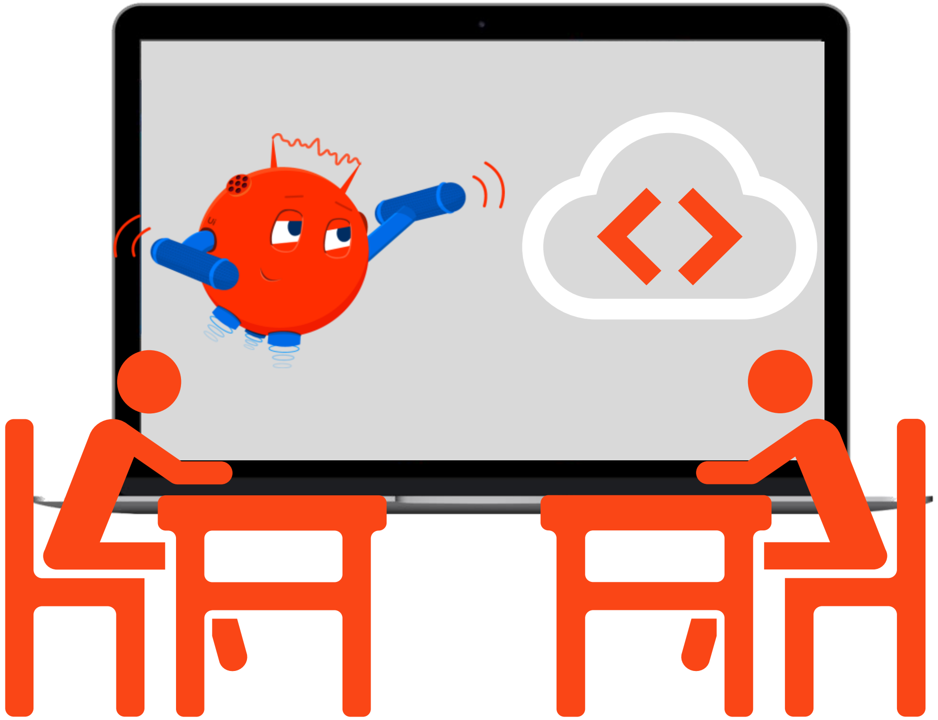
Switch Activity

Classroom Exercise

Demonstrate the use of **Switch** activity by building a workflow that does the following:

- Takes numbers between 1 and 7 as input from the user. Here, 1 is for Monday, 2 for Tuesday, and so on till Sunday
- Checks the input number against the defined condition:
  - If the number entered is between 1 and 5, then it displays on the screen “It’s a weekday” in a message box
  - If the number entered is 6 or 7, then it displays on the screen “It’s weekend” in a message box
  - If the number entered is not between 1 and 7 then it displays an error saying, “Invalid entry, please enter the number between 1 and 7 only”

Practice Exercise – Switch Activity



Build a workflow using **Switch** activity that asks user their eye color and display their personality type in a message box.

- Ask the user for their eye color
- If the user chooses “Blue”, respond with “You must be very Brave!”
- If the user chooses “Green”, respond with “You must be Generous!”
- If the user chooses “Gray”, respond with “You must be very Wise!”
- If the user chooses “Black”, respond with “You must be very Bold!”
- If the user chooses “Other”, respond with “Your eyes are unique!”

If Activity vs. Switch Activity

If Activity

- Suitable for taking a decision
- Checks for equality and logical expression
- Expression decides whether the statements inside the ‘Then’ or ‘Else’ section is to be executed

Switch Activity

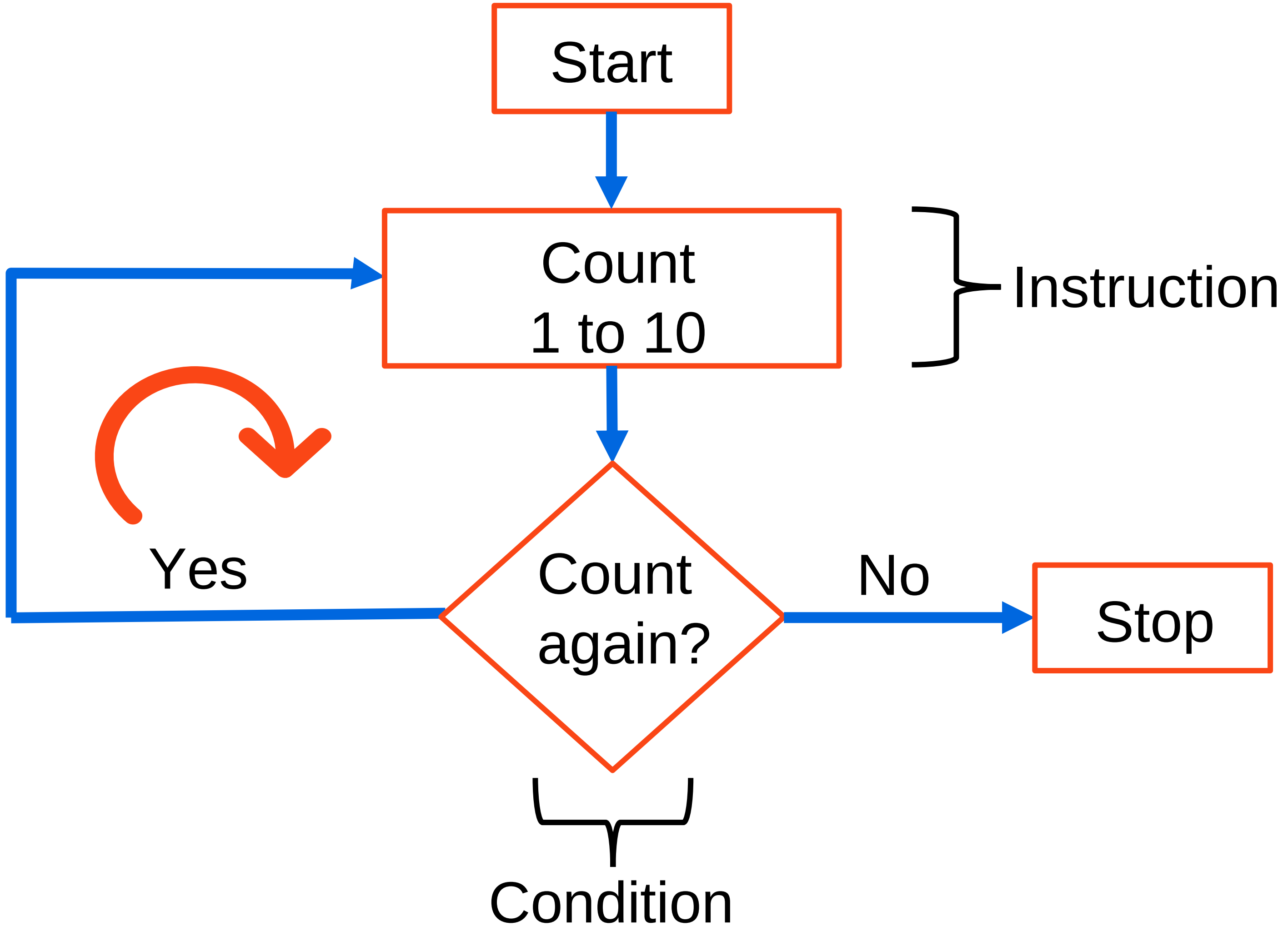
- Suitable to test the value of a given variable against a list of case value
- Checks only for equality
- Expression decides which case to execute

Loops

- Introduction to Loops
- Types of Loop Activities
  - Do While
  - While
  - For Each

Introduction to Loops

Loops repeat a sequence of instructions until a specific condition is met or till the condition is valid.



Types of Loop Activities

There are three types of loop activities available in Studio:

D

Do While

W

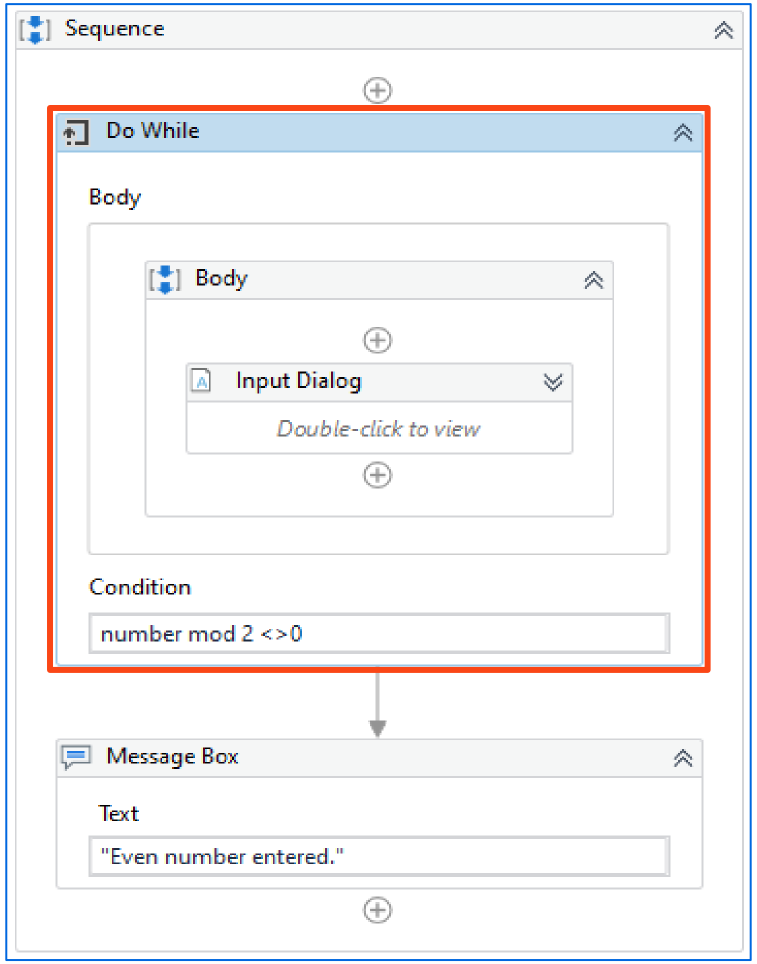
While

F

For Each

Do While Activity

The Do While activity enables the user to execute a specified part of the automation project while a condition is met. The project exits the loop when the specified condition is no longer met.



number mod 2 <>0

"Even number entered."

UiPath™ Academy

Chapter 4:

Control Flow in Studio

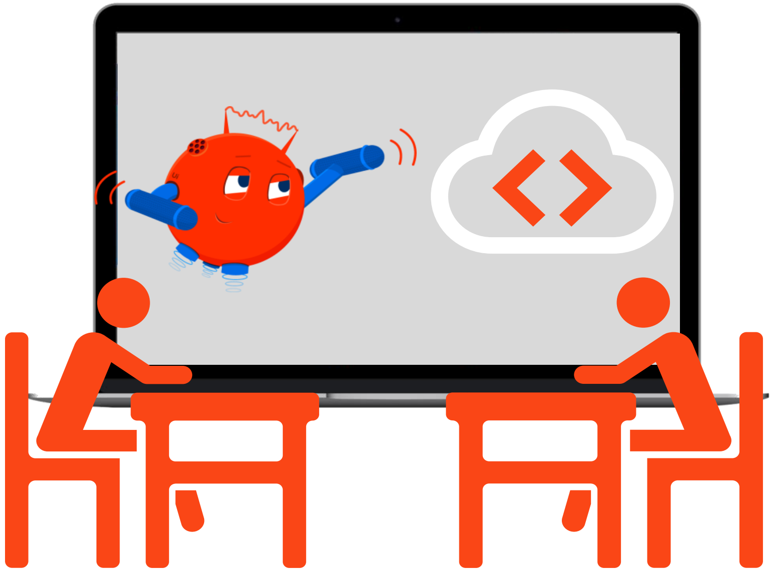
Do While

Classroom Exercise

Demonstrate the use of **Do While ac**tivity by building a workflow that asks the user to input an even number. The program continues to ask for the input until an even number is entered.

- Ask for an even number input from the user
- If the user enters an odd number:
  - Display "It's an odd number. Try again."
  - Again, ask for an even number input
- If the user enters an even number:
  - Display "Right! It's an even number."
  - End the program

Practice Exercise – Do While

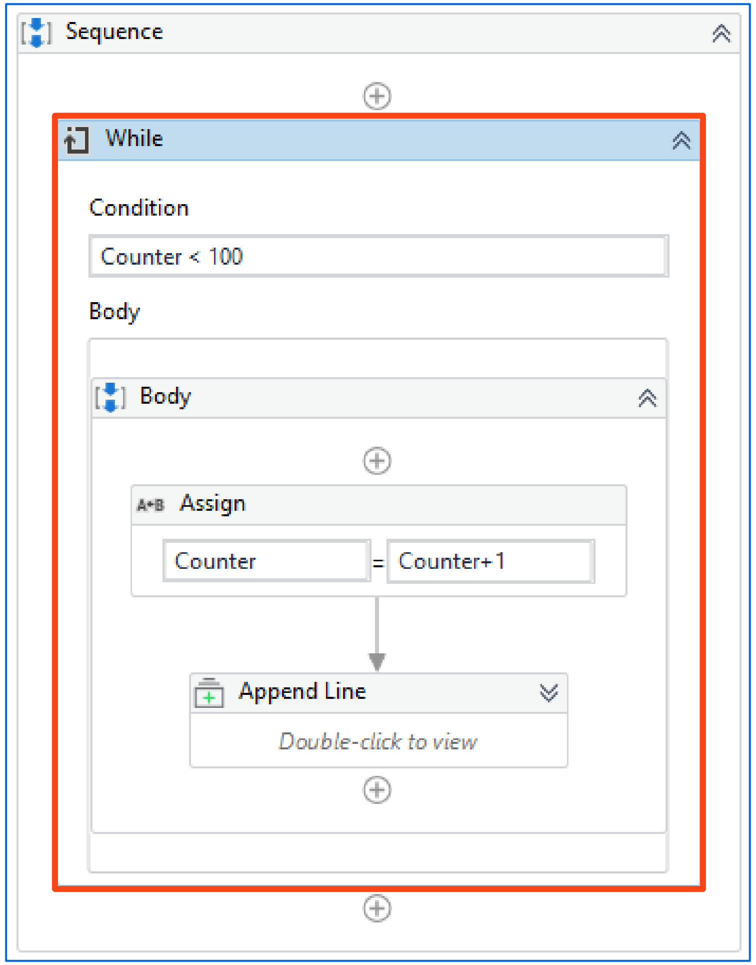


Build a workflow using a **Do While** activity for creating a 'Guessing Game' with the following conditions:

1. Generate a random number and prompt the user to input a number
2. If the input number is greater than the number generated, then it should display the message: 'Please enter a lesser number'
3. If the input number is lesser than the number generated, then it should display the message: 'Please enter a greater number'
4. The loop keeps on running until the input number equals to the generated number

While Activity

While activity enables the user to create a loop that executes the contained steps repeatedly, while a specific condition is met. The condition is evaluated before each execution of the sequence.



Condition

Counter < 100

Body

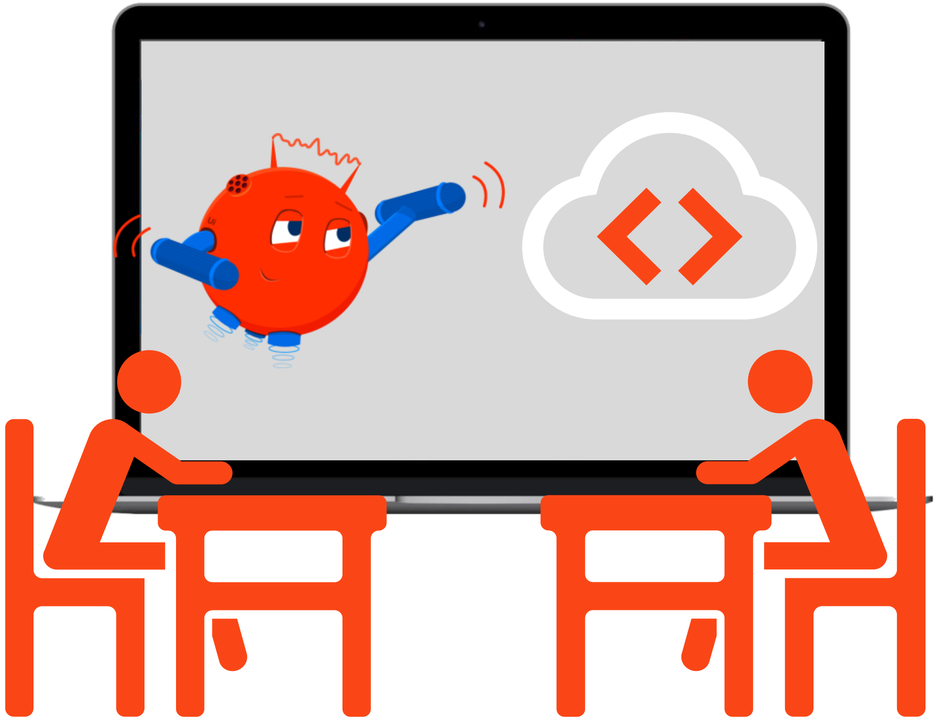
Counter = Counter+1

Classroom Exercise

Demonstrate the use of **While** activity to print a list of numbers.

- Ask the user to input a number greater than 1
- Print a list of prime numbers between 1 and the number given by the user

Practice Exercise – While Activity

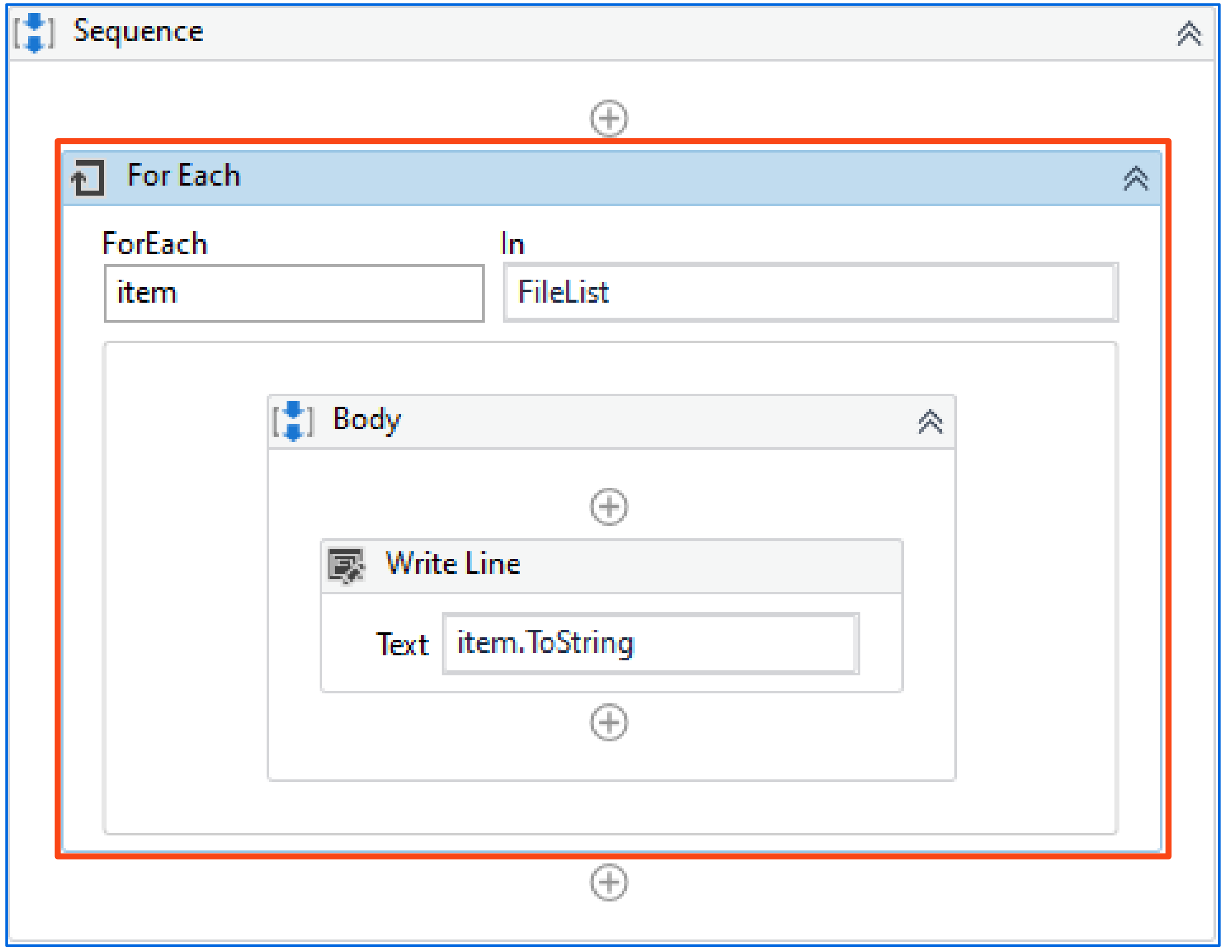


Build a workflow using a **While** activity which informs the user whether the input is a prime number.

- 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

For Each Activity

For Each activity enables the user to step through arrays, lists, data tables or other types of collections, so that the user can iterate through the data and process each piece of information individually.



ForEach: item

In: FileList

Text: item.ToString

UiPath™ Academy

Chapter 4:

Control Flow in Studio

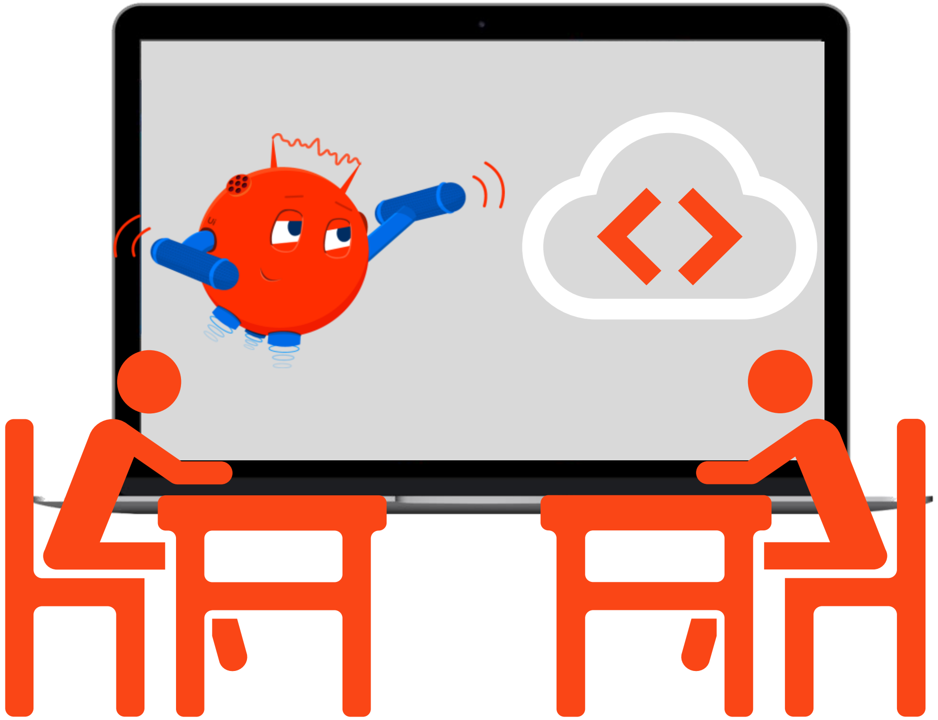
For Each

Classroom Exercise - For Each activity

Demonstrate the use of **For Each** activity. Build a workflow that displays the directory name of all the files from a folder.

- Locate and select a folder containing multiple files
- List the directory path of all the files in the Output panel

Practice Exercise - For Each activity



Build a workflow using **For Each** activity 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

Other Control Flow Activities

- Delay
- Break
- Assign
- Continue
- Parallel

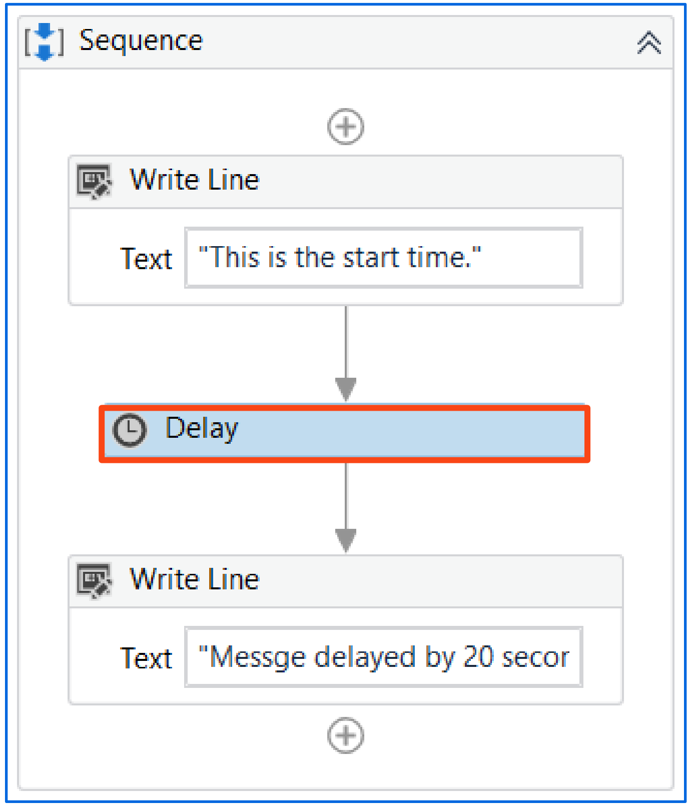
Other Control Flow Activities

In addition to decision-based and iteration-based control flow activities, there are other activities in Studio that help in controlling the flow of the program. These are:

- Delay
- Break
- Assign
- Continue
- Parallel

Delay Activity

Pauses the automation for the duration as defined by the user



Sequence

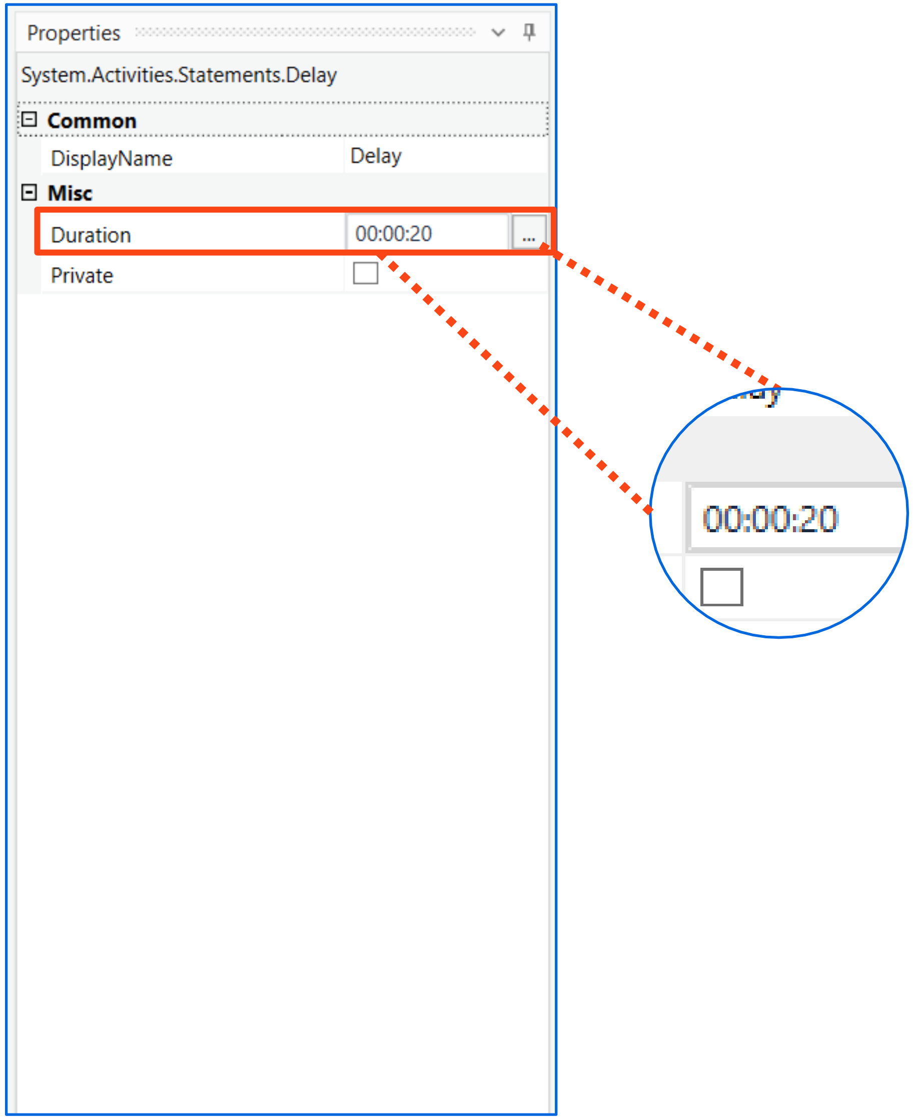
Write Line

Text "This is the start time."

Delay

Write Line

Text "Messge delayed by 20 secor



Properties

System.Activities.Statements.Delay

⊟ **Common**

DisplayName Delay

⊟ **Misc**

Duration 00:00:20

Private □

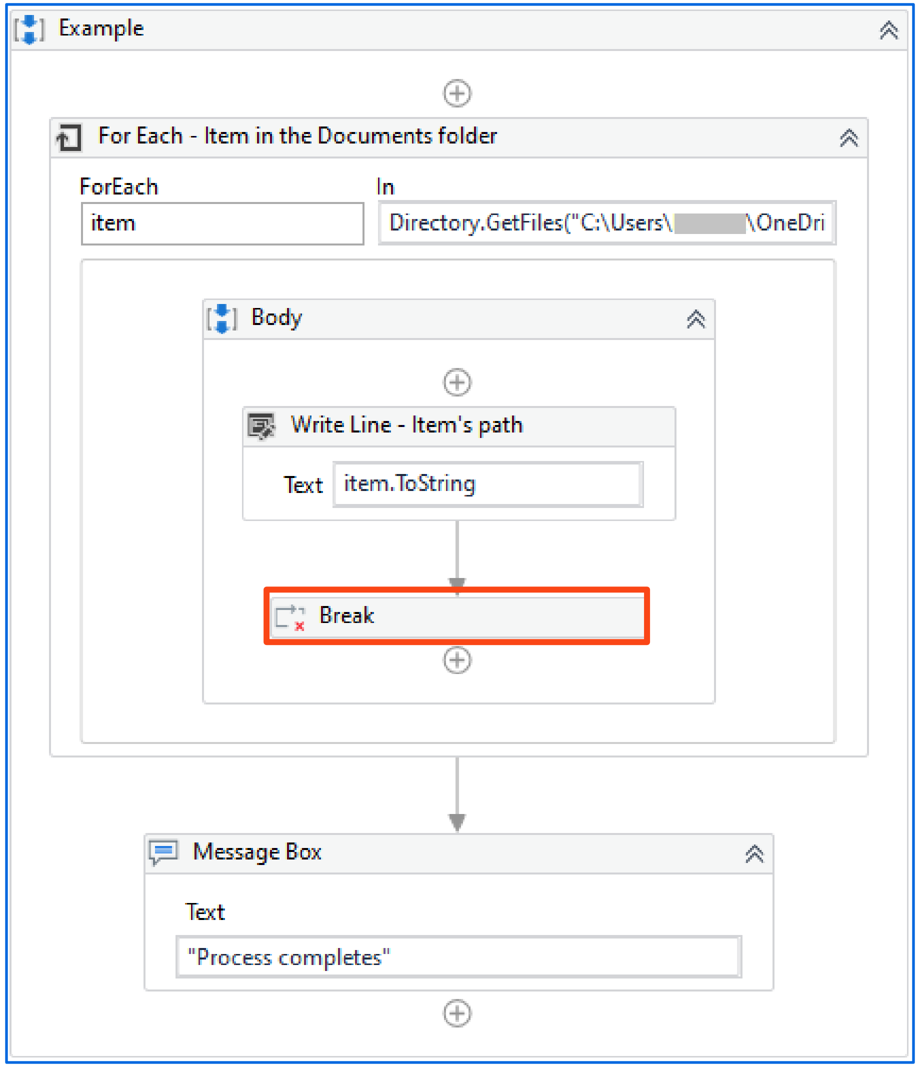
Classroom Exercise - Delay

Demonstrate the use of **Delay** activity to delay a process by a certain amount of time.

- Ask the user to open a notepad within 30 seconds
- Pause the process for 30 seconds
- Write a piece of text in the opened notepad

Break Activity

Stops the loop at a chosen point, and the process execution continues with the next activity.



Example

For Each - Item in the Documents folder

ForEach: item

In: Directory.GetFiles("C:\Users\[redacted]\OneDri

Body

Write Line - Item's path

Text: item.ToString

Break

Message Box

Text: "Process completes"

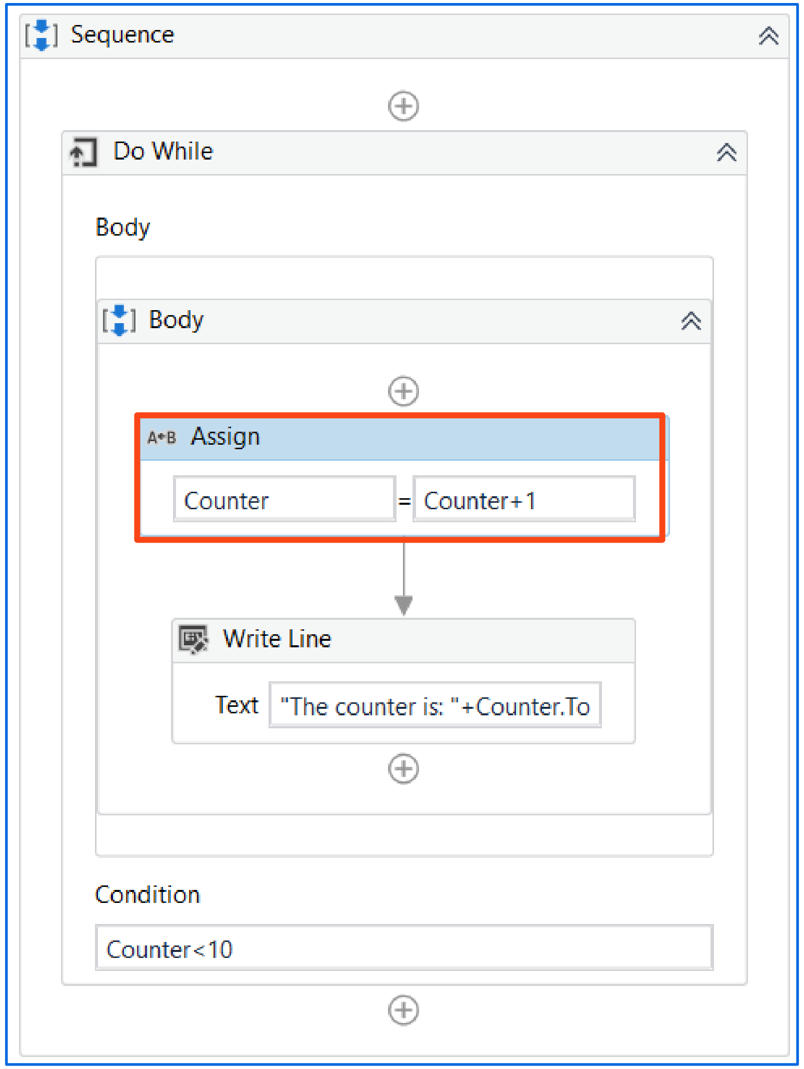
Classroom Exercise - Break

Demonstrate the use of **Break** activity to stop a process by building a workflow that does the following:

- Run a loop using the For Each activity to append a text in each item in an array of integers
- After each iteration, display a message box and ask the user "One iteration is done. Do you want to stop?
- If the user replies with ‘Yes’ then the iteration stops, else it continues
- Display the output in the Output panel

Assign Activity

Allocates a value to a variable or an argument.



Counter = Counter+1

Text "The counter is: "+Counter.To

Counter<10

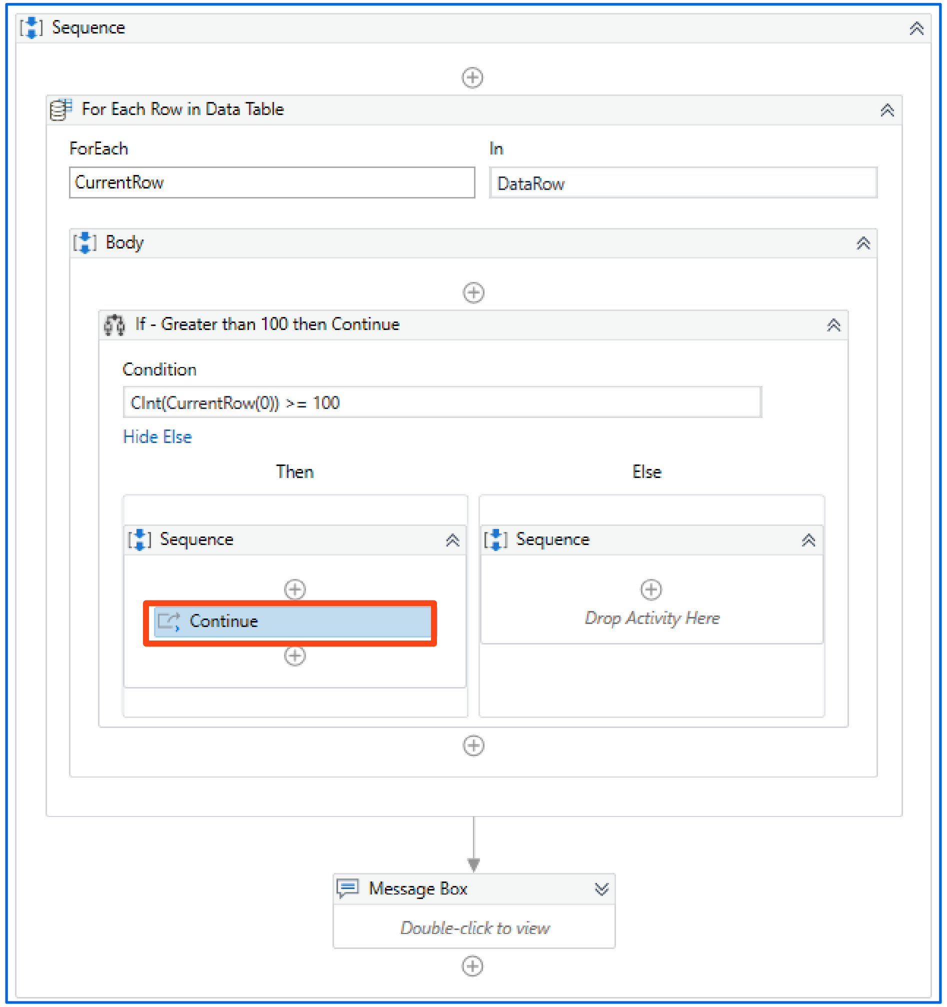
Classroom Exercise - Assign

Demonstrate the use of **Assign** activity by building a workflow in which we assign a value to a variable.

- Ask the user to input his name and then assign it to a string variable using Assign activity
- Display "Hello! How can I assist you, User". Replace "User" with user's name

Continue Activity

Enables you to skip the remaining steps in the current iteration inside a loop activity, such as For Each, While, or Do While loop.



Sequence

For Each Row in Data Table

ForEach: CurrentRow

In: DataRow

Body

If - Greater than 100 then Continue

Condition: CInt(CurrentRow(0)) >= 100

Hide Else

Then

Sequence

Continue

Else

Sequence

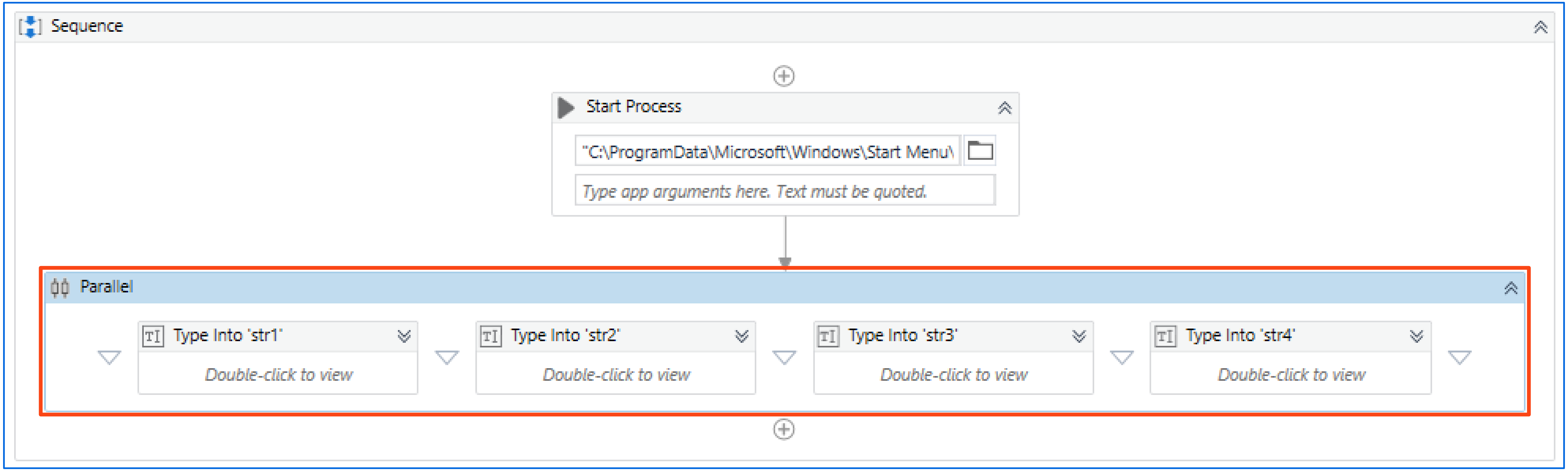
Drop Activity Here

Message Box

Double-click to view

Parallel Activity

Used to execute multiple activities asynchronously.

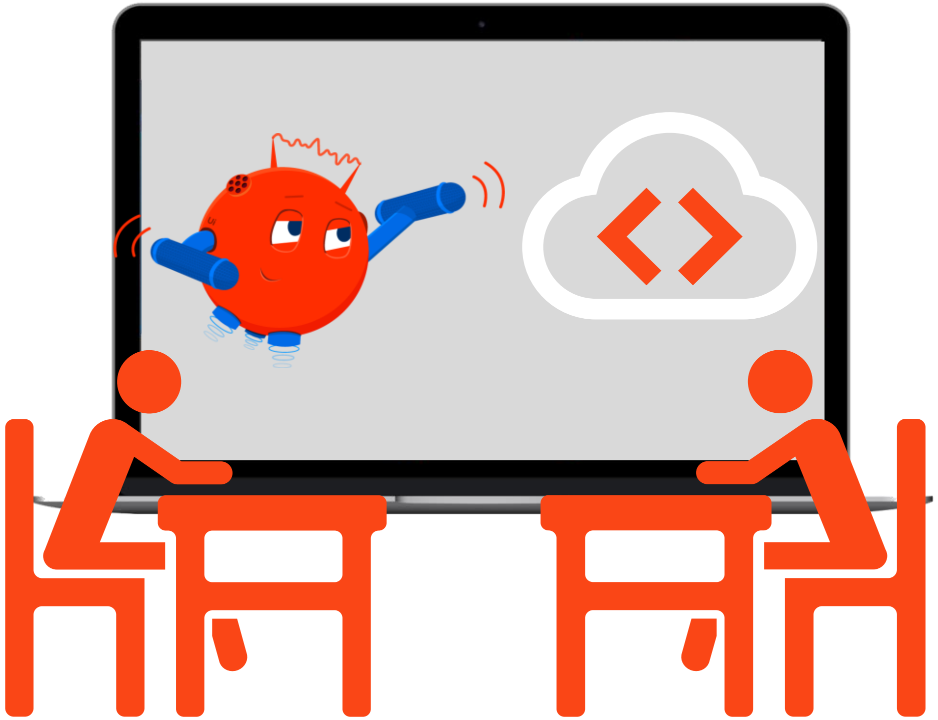


Classroom Exercise

Demonstrate the use of **Parallel** activity by running two processes in parallel and writing a piece of text in two separate notepad windows.

- Create two processes to write texts in two different notepads
- Write texts in both the notepads simultaneously

Practice Exercise - Parallel activity



Build a workflow using a **Parallel** activity to do the following:

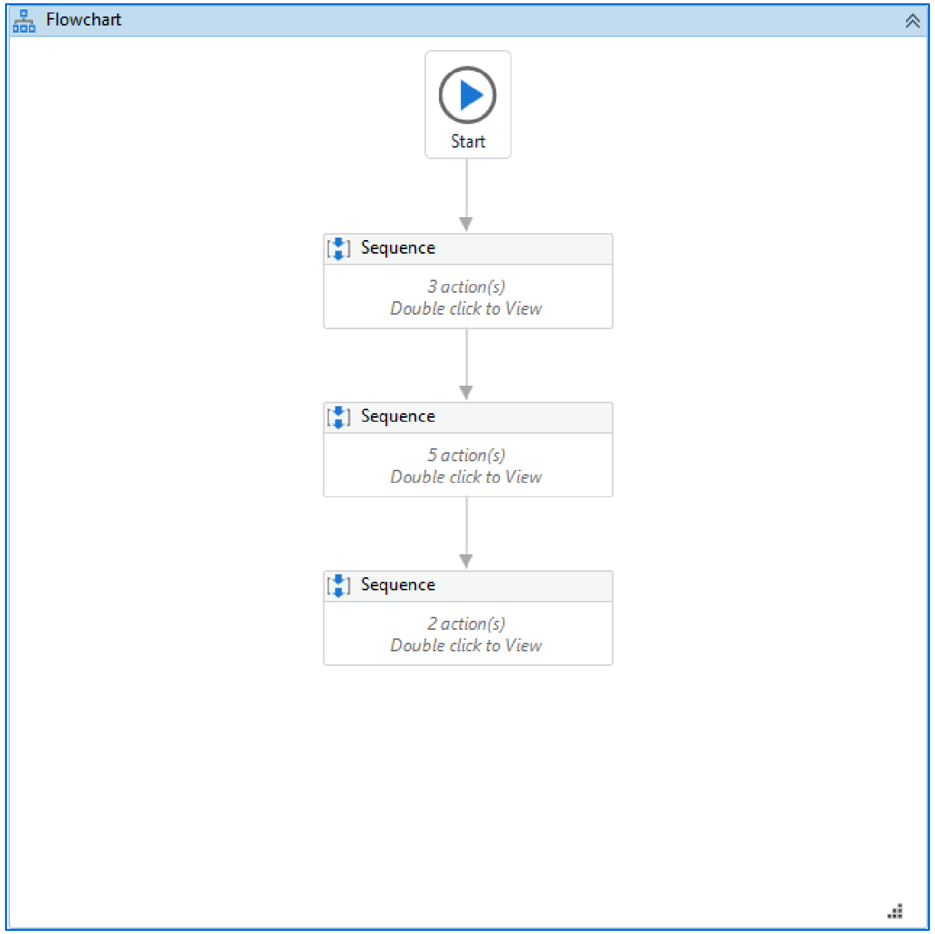
- Perform the following activities in parallel:
    - Open UiPath website, copy the text from the ‘What is Robotic Process Automation?’ section
    - Open UiPath website, copy the text from the ‘What is Process Mining?’ section
    - Open UiPath website, copy the text from the RPA Journey webpage
- Finally, store all copied text in an MS Word file
- Save and close the MS Word file

Flowcharts

- Introduction to Flowcharts
- Decision-Making in Flowcharts
- Loops in Flowcharts
- Nesting Flowcharts and Sequences
- Sequences vs. Flowcharts

Introduction to Flowcharts

Flowcharts consist of various activities which can be connected to one another in multiple ways and diagrammatically represent various steps involved in completing activities, tasks, and processes.



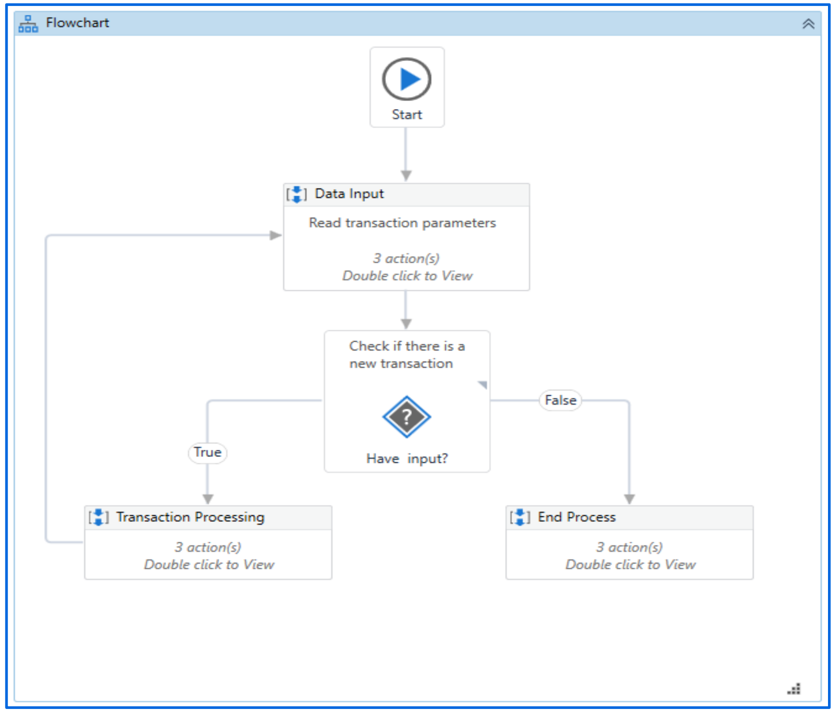
Classroom Exercise - Flowcharts

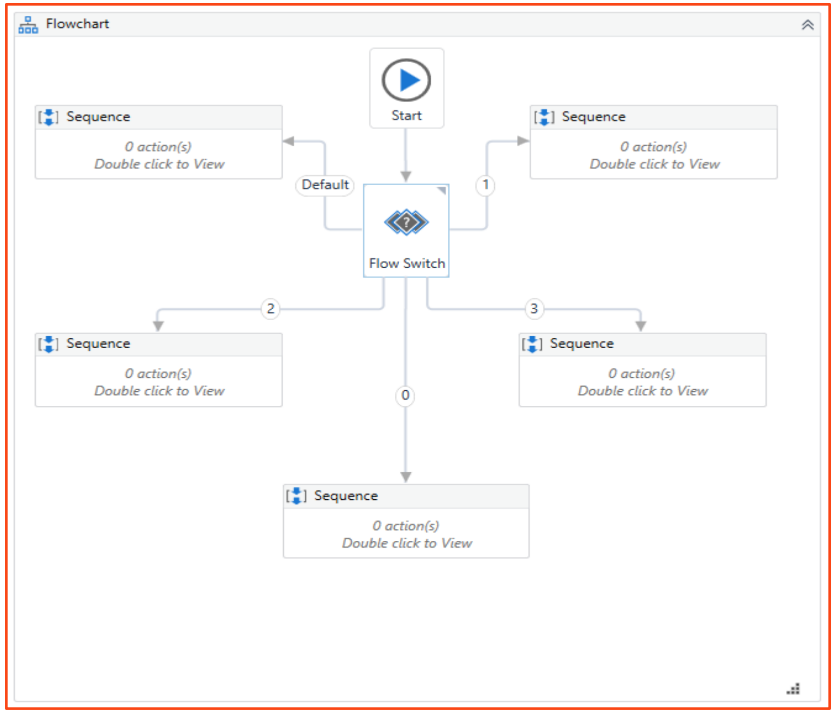
Demonstrate the use of **Flowcharts** by building a workflow that informs the user whether they have passed the exam.

- Take input from a user of name & marks obtained in an exam
- Check if the marks are greater than 60
- Display the result in a message box for Pass as “Congratulations! you have passed the exam.”
- Display the result in a message box for Fail as “Hello User, you have failed the exam.” Replace "User" with the user's name

Decision-Making in Flowcharts

There are two activities used to decide the flow of a program inside a flowchart:





- **Flow Decision**
  (equivalent to the If activity)

- **Flow Switch**
  (equivalent to the Switch activity)

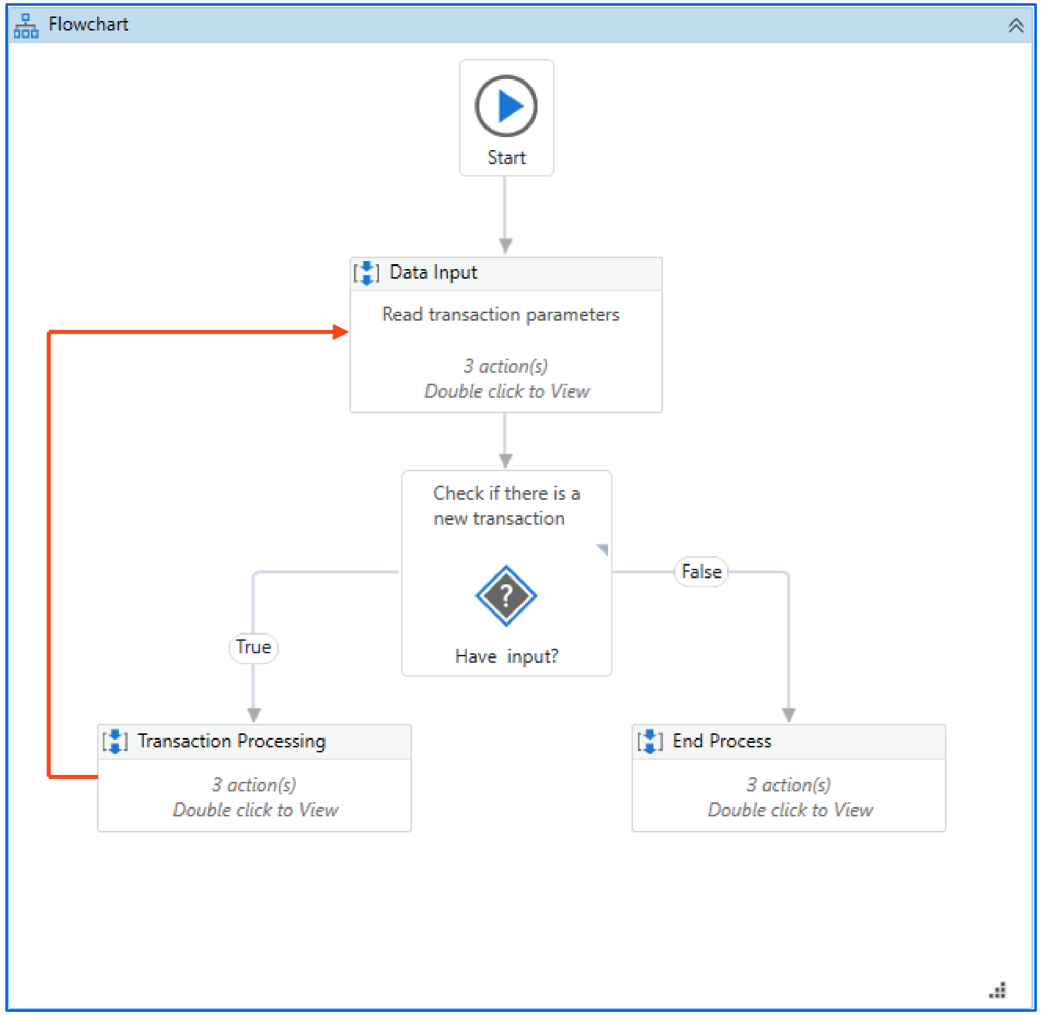
Classroom Exercise - Decisions in Flowcharts

Demonstrate the use of **Decisions in Flowcharts** by building a workflow that informs the user if they are eligible to vote.

- Ask the user to input his age
- Voting age should be greater than or equal to 18
- Display “You can vote!” in a message box if the input age is greater than 18
- Display “You cannot vote!” in a message box if the input age is less than 18
- Use a Flow Decision activity to decide whether the user can vote

Loops in Flowcharts

Used when a certain part of the flowchart is required to perform a repetitive task based on a specified condition.

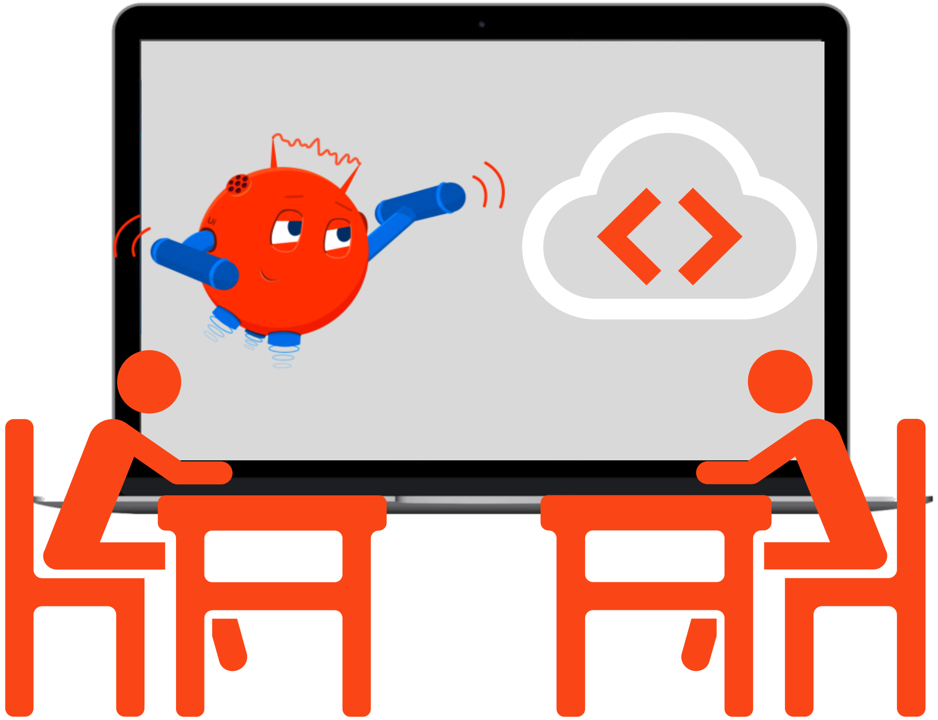


Classroom Exercise

Demonstrate the use of **Loop in Flowchart** by building a workflow that informs the user if the input is an even number.

- Ask the user to enter an even number
- Continue asking for input until the number entered is an even number
- If the number entered is an even number, then display in a message box “Perfect!”
- If the number entered is an odd number, then display in a message box “Try again.”

Practice Exercise - Loop in Flowchart



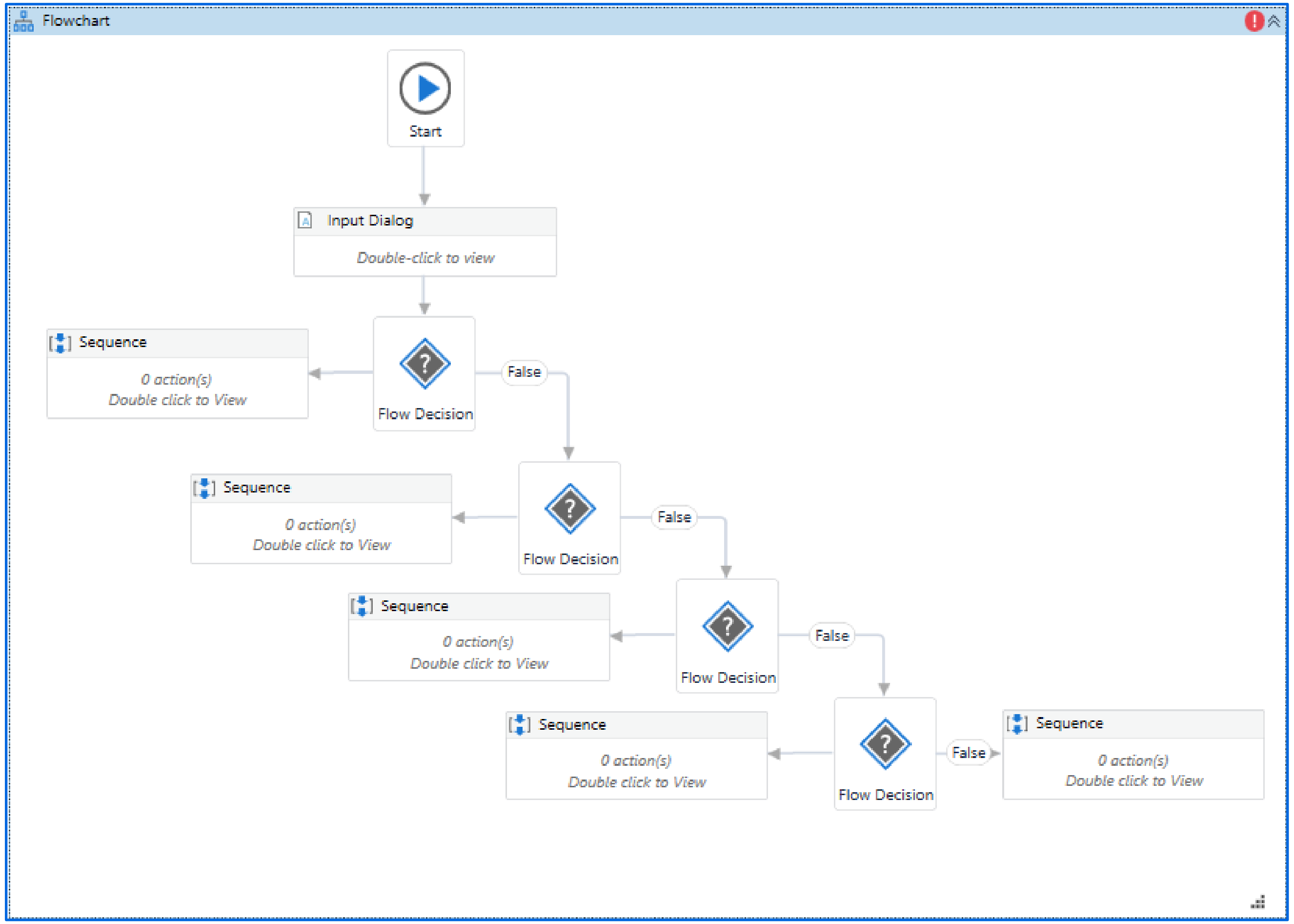
Build a workflow using **Loop in Flowchart** that asks the user for their name and two-digit lottery number and displays if they are a winner.

- Ask the name of the user and two-digit lottery number
- Give the user five chances to enter the correct lottery number
- If the number entered is below 54 and above 64, display "Enter your lottery number. X chance remaining." Here, X is the number of remaining chances for the user
- If the number entered is between 54 and 64, display, "Congratulations User! You won the lottery." Replace User with the name of the user
- If chances end before correct entry, display "Sorry, you lost. No more chances are remaining."

Nesting Flowcharts and Sequences

Nesting is used while creating complex workflows, as it allows a logical division of the program and promotes reusability.

Sequences nested within a flowchart activity



Sequences vs. Flowcharts

The differences between sequences and flowcharts are:

Sequences	Flowcharts
Suitable for linear processes	Suitable for complex processes
Seamless movement from one activity to another	Complex movement between two activities, but easier to understand as a whole
Act as a single block of activity	Connect multiple block of activities through branching

Resources

Topic	Link
About Control Flow - Official Documentation	https://docs.uipath.com/studio/docs/about-control-flow
Control Flow Activities - Official Documentation	https://docs.uipath.com/studio/docs/control-flow-activities
Sequences - Official Documentation	https://docs.uipath.com/studio/docs/sequences
Flowcharts - Official Documentation	https://docs.uipath.com/studio/docs/flowcharts
State Machines - Official Documentation	https://docs.uipath.com/studio/docs/state-machines

Resources

Topic	Link
Control Flow Activities - Official Documentation	https://docs.uipath.com/studio/docs/control-flow-activities
If Activity - Official Documentation	https://docs.uipath.com/activities/docs/if
Else If Activity - Official Documentation	https://docs.uipath.com/activities/docs/if-else-if
Flow Decision - Official Documentation	https://docs.uipath.com/activities/docs/flow-decision
If Operator (Visual Basic) - Microsoft .Net Documentation	https://learn.microsoft.com/en-us/dotnet/visual-basic/language-reference/operators/if-operator

Resources

Topic	Link
Do While - Official Documentation	https://docs.uipath.com/activities/docs/interruptible-do-while
While - Official Documentation	https://docs.uipath.com/activities/docs/interruptible-while
For Each - Official Documentation	https://docs.uipath.com/activities/docs/for-each
Break/ Exit Loop - Official Documentation	https://docs.uipath.com/activities/docs/break
Break Activity - Official Documentation	https://docs.uipath.com/studio/docs/the-break-activity

Resources

Topic	Link
Switch - Official Documentation	https://docs.uipath.com/activities/docs/switch
Flow Switch - Official Documentation	https://docs.uipath.com/activities/docs/flow-switch
Directory Class - Microsoft .Net Documentation	https://learn.microsoft.com/en-us/dotnet/api/system.io.directory?view=netframework-4.8
String Functions - Microsoft .Net Documentation	https://learn.microsoft.com/en-us/dotnet/visual-basic/language-reference/functions/string-functions