Unit 2: Electronic Spreadsheet (Advanced)
Chapter 5: Using Macros in Spreadsheet

Macro: An Overview

A **macro** in LibreOffice Calc is a series of commands and functions stored in a script, allowing users to **automate repetitive tasks**.

1. Advantages of Macros

- **Automation** of repetitive tasks (e.g., formatting, data entry).
- **Time-saving** by executing multiple steps in one click.
- **Improves accuracy** by minimizing manual input.
- **Customizes** spreadsheet behavior for specific needs.

2. Rules for Naming a Macro

- Macro names **must begin with a letter**.
- They can include **letters, numbers, and underscores (_)**.
- **Spaces and special characters** (like %, #, @) are not allowed.
- Macro names are **case-insensitive** (e.g., TotalCalc and totalcalc are the same).

3. Using the Macro Recorder

The Macro Recorder allows users to **record actions** without writing code.
Tools→ option → Libra office→advanced→check enable macro recording→ok

4. Recording Macros

Steps:

1. Go to **Tools** > **Macros** > **Record Macro**.
2. Perform the desired actions (e.g., formatting, entering data).
3. Click **Stop Recording** when done.
4. In the dialog box:
    - Choose where to save the macro (**My Macros** or current document).
    - Name the macro and click **Save**.

Note: The macro recorder only records user-interface actions, not complex logic.

5. Running a Macro

Steps:

1. Go to **Tools** > **Macros** > **Run Macro**.
2. In the dialog, expand the macro library to find your macro.
3. Select the macro and click **Run**.

What is a Library in Macros?

A **Library** in LibreOffice Macros is like a **folder** that holds multiple **modules** or **dialogs**. Each module contains your **macro code**, written in **LibreOffice Basic**, **Python**, or another supported language.

Structure:

- **Library**
    - **Module 1** (contains macro code)
    - **Module 2**
    - **Dialog 1** (for user interfaces, if any)

6. Organizing a Macro

6.1 To Create or Edit a Macro Using Functions

You can use the **LibreOffice Basic editor** to write or modify macros.

Steps:

1. Go to **Tools** > **Macros** > **Organize Macros** > **Basic**.
2. Click **Edit** to open the editor.
3. Write or modify code using LibreOffice Basic.

Example:

```
Function HelloWorld()
   MsgBox "Hello, this is a macro!"
End Function
```

6.2 Storing a Macro Internally

Macros can be stored in:

- **My Macros** – Available for all documents.
- **Current Document** – Stored inside the .ods file.
- **Application Macros** – Accessible system-wide.

Steps:

- When saving or creating a macro, choose the appropriate location in the macro dialog box.

7. Comparison of Predefined Functions and Macros in Spreadsheet Application

Feature	Predefined Functions	Macros
Built-in availability	Yes	No, must be created
Complexity	Limited to cell formulas	Can handle complex tasks
Customization	Limited	Highly customizable
Execution	Automatically in cell	Requires manual execution or trigger
Use Case	Calculations	Automation, workflows, customization

8. Passing Argument to a Macro

Macros can be designed to accept input (arguments) for dynamic behavior.

Example:

```
Function GreetUser(name)
   MsgBox "Hello, " & name
End Function
```

9. Argument Passed as Values

You can pass literal values when calling a macro.

Example:

=GreetUser(“Satyendra”)

This will display: **Hello, Satyendra**

You can also pass **cell values**:

10. Macros Act Like Built-in Functions

You can define macros that act like functions in formulas.

Example:

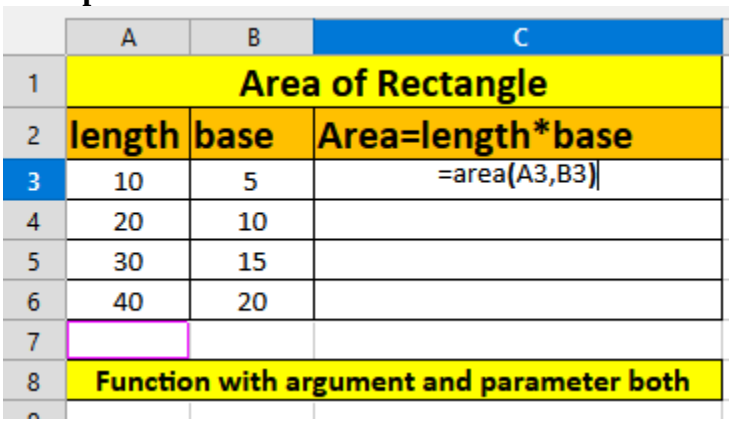
```
Function AddTwoNumbers(a, b)
   AddTwoNumbers = a + b
End Function
```

Usage in Calc:

=AddTwoNumbers(5,10)

11. Sorting Columns Using Macro

Example Macro to Sort a Column:



	A	B	C
1	Area of Rectangle		
2	length	base	Area=length*base
3	10	5	=area(A3,B3)
4	20	10	
5	30	15	
6	40	20	
7			
8	Function with argument and parameter both		