**Module 1: Overview of the Basics of Excel**

**1. Customizing Common Options**

*No data required — conceptual.*

1. Add “New Sheet” and “Sort Ascending” commands to your **Quick Access Toolbar**.
2. Go to **File → Options → General** and change the **default number of worksheets** in a new workbook from 1 to 3.

**2. Absolute and Relative Cells**

| **Product** | **Price** | **Quantity** | **Total** |
| --- | --- | --- | --- |
| Pen | 10 | 5 |  |
| Pencil | 5 | 10 |  |
| Marker | 25 | 3 |  |

**Questions:**

1. In cell **D2**, type the formula =B2\*C2 and copy it down. Observe how relative references work.
2. In a new column, calculate **Total with Tax** using an **absolute cell** reference. Assume **Tax Rate = 10%** stored in cell **F1**.  
   Formula in D2: =B2\*C2\*$F$1

**3. Protecting and Unprotecting Worksheets**

| **Name** | **Salary** | **Bonus** |
| --- | --- | --- |
| Amit | 40000 | 4000 |
| Priya | 35000 | 3500 |
| Swapnil | 30000 | 3000 |

**Task:**

1. Lock only the **Salary** column and protect the sheet so others cannot edit it.
2. Then unprotect it to allow edits again.

**Module 2: Working with Functions**

**1. IF Function (Conditional Expressions)**

| **Student** | **Marks** |
| --- | --- |
| Ravi | 85 |
| Neha | 45 |
| Suresh | 60 |

**Tasks:**

1. Write a formula in column **C**:  
   =IF(B2>=50, "Pass", "Fail")
2. Extend the formula to assign grades:  
   =IF(B2>=80,"A",IF(B2>=60,"B",IF(B2>=50,"C","Fail")))

**2. Logical Functions (AND, OR, NOT)**

| **Name** | **Age** | **Experience (Years)** |
| --- | --- | --- |
| Amit | 28 | 5 |
| Neha | 32 | 2 |
| Rahul | 24 | 4 |

**Tasks:**

1. Check if an employee is **eligible** (Age > 25 AND Experience ≥ 3):  
   =IF(AND(B2>25,C2>=3),"Eligible","Not Eligible")
2. Use OR function:  
   =IF(OR(B2>30,C2>4),"Consider","Reject")

**3. Lookup and Reference (VLOOKUP, HLOOKUP, MATCH, INDEX)**

**Data Table (Products)**

| **Product ID** | **Product Name** | **Price** |
| --- | --- | --- |
| P101 | Pen | 10 |
| P102 | Pencil | 5 |
| P103 | Marker | 25 |

**Task Table**

| **Lookup ID** | **Product Name (using VLOOKUP)** |
| --- | --- |
| P102 |  |

**Tasks:**

1. Find product name using:  
   =VLOOKUP(A2,Products!A:C,2,FALSE)
2. Use **INDEX + MATCH** to find Price:  
   =INDEX(C2:C4, MATCH(A2,A2:A4,0))

**4. VLOOKUP with Exact & Approximate Match**

| **Marks** | **Grade** |
| --- | --- |
| 0 | F |
| 50 | D |
| 60 | C |
| 70 | B |
| 85 | A |

**Task Table**

| **Student** | **Score** | **Grade** |
| --- | --- | --- |
| Amit | 78 |  |
| Neha | 63 |  |

**Tasks:**

1. Use Approximate Match to assign grades:  
   =VLOOKUP(B2,$A$2:$B$6,2,TRUE)
2. Change the 4th argument to FALSE and see what happens with an **Exact Match**.

**5. Nested VLOOKUP (Exact Match)**

**Table 1 – Employee**

| **Emp ID** | **Dept Code** |
| --- | --- |
| E001 | D01 |
| E002 | D02 |
| E003 | D03 |

**Table 2 – Department**

| **Dept Code** | **Department Name** |
| --- | --- |
| D01 | HR |
| D02 | IT |
| D03 | Finance |

**Task:**

1. Write a **nested VLOOKUP** to get Department Name using Emp ID:  
   =VLOOKUP(VLOOKUP(A2,Table1!A:B,2,FALSE),Table2!A:B,2,FALSE)

**6. VLOOKUP with Tables / Dynamic Ranges**

| **ID** | **Name** | **Salary** |
| --- | --- | --- |
| 101 | Ravi | 30000 |
| 102 | Neha | 40000 |
| 103 | Amit | 50000 |

**Tasks:**

1. Convert the data into a **Table** (Ctrl + T) named EmpTable.
2. Use formula:  
   =VLOOKUP(102,EmpTable,3,FALSE)
3. Add a new row and see how the dynamic range automatically expands.

**7. VLOOKUP to Consolidate Data from Multiple Sheets**

**Sheet1 (North Region)**

| **ID** | **Sales** |
| --- | --- |
| N101 | 45000 |
| N102 | 32000 |

**Sheet2 (South Region)**

| **ID** | **Sales** |
| --- | --- |
| S101 | 28000 |
| S102 | 36000 |

**Summary Sheet**

| **Region** | **ID** | **Sales** |
| --- | --- | --- |
| North | N101 |  |
| South | S102 |  |

**Tasks:**

1. Use VLOOKUP with **INDIRECT** to pull data from different sheets:  
   =VLOOKUP(B2,INDIRECT(A2&"!A:B"),2,FALSE)
2. Change Region to “North” or “South” and see the result update automatically.

**Module 3: Data Validation**

**1. Specifying a valid range of values for a cell**

| **Employee** | **Age** |
| --- | --- |
| Amit |  |
| Neha |  |
| Rahul |  |

**Tasks:**

1. Apply **Data Validation** on column **B (Age)** to allow only numbers between **18 and 60**.  
   → Use: *Data → Data Validation → Whole number → between 18 and 60*
2. Try entering 15 or 70 and note the validation message.

**2. Specifying a list of valid values for a cell**

| **Employee** | **Department** |
| --- | --- |
| Amit |  |
| Neha |  |
| Rahul |  |

**Tasks:**

1. Create a dropdown list for **Department** using these options:  
   **HR, IT, Finance, Sales, Admin**
2. Add input message: *“Select department from list only.”*
3. Add an error alert if someone types a non-listed department.

**3. Specifying custom validations based on formula**

| **Name** | **Start Date** | **End Date** |
| --- | --- | --- |
| Project A | 01-Jan-2025 |  |
| Project B | 15-Feb-2025 |  |

**Tasks:**

1. Ensure **End Date** is always greater than **Start Date** using:
   * Formula in Data Validation → Custom:  
     =C2>B2
2. Try entering an End Date earlier than Start Date to test the restriction.
3. Add a custom error message like *“End Date must be after Start Date.”*

**Module 4: Working with Templates**

**1. Designing the structure of a template**

| **Header** | **Details** |
| --- | --- |
| Company Name |  |
| Report Date |  |
| Prepared By |  |
| Department |  |
| Total Employees |  |

**Tasks:**

1. Design a **Company Report Template** layout:
   * Include a header section (above table)
   * Add sections for **Employee Data**, **Department Summary**, etc.
2. Save the file as a **Template (.xltx)**:
   * Go to *File → Save As → Excel Template (.xltx)*

**2. Using templates for standardization of worksheets**

| **Name** | **Department** | **Salary** | **Bonus** |
| --- | --- | --- | --- |
| Amit | IT | 50000 |  |
| Priya | HR | 45000 |  |
| Rahul | Finance | 55000 |  |

**Tasks:**

1. Create a **Payroll Template**:
   * Predefine columns and formatting
   * Include formulas like =C2\*0.1 for Bonus (10%)
2. Save it as “Payroll\_Template.xltx”
3. Open a **new workbook using this template** and fill fresh data for a new month.

**Module 5: Sorting and Filtering Data**

**1. Sorting Tables**

| **Name** | **Department** | **Salary** |
| --- | --- | --- |
| Amit | IT | 50000 |
| Priya | HR | 40000 |
| Neha | Sales | 45000 |
| Rahul | Finance | 55000 |

**Tasks:**

1. Sort the data **alphabetically by Name** (A–Z).
2. Sort again by **Salary (Highest to Lowest)**.

**2. Using Multiple-Level Sorting**

| **Product** | **Category** | **Sales** |
| --- | --- | --- |
| Pen | Stationery | 5000 |
| Pencil | Stationery | 3000 |
| Keyboard | Electronics | 10000 |
| Mouse | Electronics | 8000 |
| Notebook | Stationery | 7000 |

**Tasks:**

1. Sort first by **Category (A–Z)**, then by **Sales (Largest to Smallest)** within each category.
2. Observe how Excel groups items by category and orders them by sales.

**3. Using Custom Sorting**

| **Month** | **Sales** |
| --- | --- |
| March | 12000 |
| January | 8000 |
| February | 10000 |
| April | 15000 |

**Tasks:**

1. Apply a **custom sort order**:  
   Order by **Month (Jan, Feb, Mar, Apr, May...)** instead of alphabetical.  
   → Use: *Sort → Custom List → Add your month sequence*
2. Verify that “January” now appears first and “April” last.

**4. Filtering Data for Selected View (AutoFilter)**

| **Name** | **Department** | **Salary** |
| --- | --- | --- |
| Amit | IT | 50000 |
| Neha | Sales | 40000 |
| Rahul | Finance | 55000 |
| Priya | HR | 42000 |

**Tasks:**

1. Apply an **AutoFilter** to show only employees in the **IT** department.
2. Filter again to show those with **Salary > 45000**.

**5. Using Advanced Filter Options**

| **Employee** | **Department** | **Salary** | **Location** |
| --- | --- | --- | --- |
| Amit | IT | 50000 | Pune |
| Neha | HR | 42000 | Mumbai |
| Rahul | Finance | 55000 | Pune |
| Priya | IT | 48000 | Delhi |
| Suresh | Sales | 39000 | Pune |

**Tasks:**

1. Create a **criteria range** like this (somewhere above or aside):

| **Department** | **Location** |
| --- | --- |
| IT | Pune |

1. Use **Advanced Filter** to extract only employees from the IT department located in Pune.
2. Copy the filtered data to a new sheet called “Filtered\_IT\_Pune.”