**Excel Assignment – 17**

1. What are modules in VBA and describe in detail the importance of creating a module?

We write VBA codes in VBA modules to perform our actions in Excel. We then execute this macro in various ways. VBA modules are stored in an Excel workbook. You can store any number of VBA modules in a workbook.

Types of VBA Modules

There are different types of VBA modules. They are:

Standard Module: This type of module mostly holds the VBA codes. Customized codes are mainly written in this type of module. The standard modules are located in the module folder in the workbook.

Sheet Module: This type of module holds code that is specially written for that specific sheet. Every worksheet in an Excel workbook has its own sheet module.

Workbook Module: This module is named ThisWorkbook by default. Every Excel workbook has only one workbook module. This type of module holds codes that are for workbook events.

UserForm Module: UserForm helps to create a customized dialog box. UserForm module holds code for that specific UserForm. An Excel workbook can contain multiple UserForm modules.

Class Module: This type of module helps to write codes that create objects, properties, and methods. You can create an object that does not exist in Object Library using the class module.

1. What is Class Module and what is the difference between a Class Module and a Module?

A Class Module in VBA (Visual Basic for Applications) is a special type of module used for defining custom objects with properties, methods, and events. It allows you to implement object-oriented programming concepts such as encapsulation, inheritance, and polymorphism within your VBA code.

|  |  |
| --- | --- |
| **Class Module** | **Standard Module** |
| Primarily used for defining custom objects and implementing object-oriented programming (OOP) concepts | Used for storing general procedures, functions, and variables that are not associated with a specific object |
| Integral to object-oriented programming | Does not inherently support object-oriented programming concepts. |
| Variables and procedures have a limited scope within the class and can be accessed by instances (objects) of that class | Variables and procedures have a broader scope and can be accessed globally within the project |
| Allows you to create multiple instances (objects) of the class, each with its own set of properties and behaviours. | Does not involve instance creation; procedures and variables are accessible without instantiation. |
| Supports the definition of properties, methods, and events associated with custom objects. | Contains general procedures and functions but does not define properties, methods, or events for objects. |
| Provides a more structured and object-oriented approach, which is beneficial for complex projects and scenarios where custom objects and encapsulation are required. | Offers a simpler and more straightforward approach suitable for general-purpose code that doesn't involve custom objects. |

1. What are Procedures? What is a Function Procedure and a Property

Procedure?  
In VBA (Visual Basic for Applications), a procedure is a set of one or more VBA statements that perform a specific task. Procedures are used to organize and execute code in a structured manner.

A Function procedure is a series of Visual Basic statements enclosed by the Function and End Function statements. A Function procedure is similar to a Sub procedure, but a function can also return a value.

A property procedure is a series of Visual Basic statements that manipulate a custom property on a module, class, or structure. Property procedures are also known as property accessors. Visual Basic provides for the following property procedures: A Get procedure returns the value of a property.

1. What is a sub procedure and what are all the parts of a sub procedure

and when are they used?

A Sub procedure is a series of Visual Basic statements enclosed by the Sub and End Sub statements. The Sub procedure performs a task and then returns control to the calling code, but it does not return a value to the calling code

Procedure Declaration: The declaration of a Sub Procedure begins with the Sub keyword followed by the name of the subroutine. It can include parameters that are passed to the subroutine.

Procedure Body: The body of the Sub Procedure contains the actual VBA code that defines the tasks to be performed. It is enclosed between the Sub and End Sub statements.

Parameters (Optional): Sub Procedures can have parameters that act as placeholders for values passed to the subroutine. Parameters are optional, and the number and type of parameters depend on the specific requirements of the subroutine.

Local Variables: Local variables are declared within the Sub Procedure and are only accessible within that procedure. They are used to store temporary values needed for calculations or other operations.

Comments: Comments are added using the single-quote (') symbol. They are used to document the code, providing explanations or descriptions of the purpose of specific lines or sections.

Flow Control Statements: Flow control statements, such as If...Then...Else, For...Next, Do...Loop, etc., can be used within a Sub Procedure to control the execution flow based on certain conditions or to repeat specific actions.

Error Handling (Optional): Error handling statements, such as On Error Resume Next or On Error GoTo, can be included to manage errors that may occur during the execution of the Sub Procedure.

Sub Procedures are used when you need to encapsulate a set of related actions into a single routine. They are commonly employed for code modularity, reusability, and readability.

1. How do you add comments in a VBA code? How do you add multiple

lines of comments in a VBA code?

Step 1: Click on the line where you want to insert a comment. Step 2: Type an Apostrophe(‘) at the start of a line. Step 3: Write the comment you want. Step 4: Press Enter and you fill find the comment written to be green.

The comment symbol: Apostrophe', or “REM,” has to be used on each line if the comments require more than one line. By default, the comments appear as green in the code window.