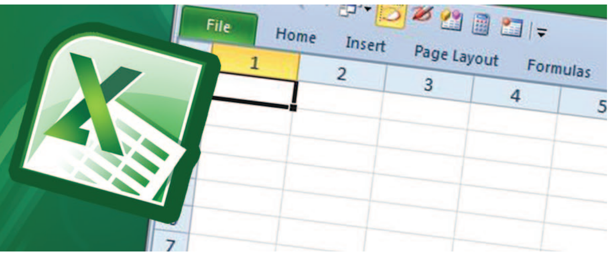
Building Blocks

Course Outline

Microsoft® Excel Power Macros



In previous courses, you used Excel to simplify business tasks, including the creation of spreadsheets, graphs, charts, and formulas that were difficult to create and nearly impossible to maintain using pencil and paper. You now want to simplify your work in the Excel environment by automating many of the repetitive tasks that are part of the spreadsheet development. In this course, you will apply Macros and the Visual Basic for Applications (VBA) programming language to simplify many of the tasks that you can perform using various tools and functions.

You will also learn the basics of VBA automation programming; learn how to employ macros with Word and Access, and benefit from numerous real-world business examples provided by the instructor.

Audience

Intermediate-level to advanced-level Excel power users, senior managers, analysts and report designers, anyone who wants to use Excel more quickly and efficiently!

Skills Taught

- Create, edit, and debug Excel macros
- Format worksheets using macros
- Create interactive Excel worksheets and forms
- Work with and process multiple worksheets using advanced automation techniques
- Perform advanced and automated calculations
- Use macros to enable/disable worksheet protection
- Create tailored power macros to speed up and enhance daily productivity

Building Blocks

Course Outline

Getting Started
- Introducing Visual Basic for Applications
- Displaying the Developer Tab in the Ribbon
- Recording a Macro
- Saving a Macro-Enabled Workbook
- Running a Macro
- Editing a Macro in the Visual Basic Editor
- Understanding the Development Environment
- Using Visual Basic Help
- Closing the Visual Basic Editor
- Understanding Macro Security

Working with Procedures and Functions
- Understanding Modules
- Creating a Standard Module
- Understanding Procedures
- Creating a Sub Procedure
- Calling Procedures
- Using the Immediate Window to Call Procedures
- Creating a Function Procedure
- Naming Procedures
- Working with the Code Editor

Understanding Objects
- Understanding Objects
- Navigating the Excel Object Hierarchy
- Understanding Collections
- Using the Object Browser
- Working with Properties
- Using the With Statement
- Working with Methods
- Creating an Event Procedure

Using Expressions, Variables, and Intrinsic Functions
- Understanding Expressions and Statements
- Declaring Variables
- Understanding Data Types
- Working with Variable Scope
- Using Intrinsic Functions
- Understanding Constants
- Using Intrinsic Constants
- Using Message Boxes
- Using Input Boxes
- Declaring and Using Object Variables

Controlling Program Execution
- Understanding Control-of-Flow Structures
- Working with Boolean Expressions
- Using the If...End If Decision Structures
- Using the Select Case...End Select Structure
- Using the Do...Loop Structure
- Using the For...To...Next Structure
- Using the For Each...Next Structure
- Guidelines for use of Control-of-Flow Structures

Working with Forms and Controls
- Understanding UserForms
- Using the Toolbox
- Working with UserForm Properties, Events, and Methods
- Understanding Controls
- Setting Control Properties in the Properties Window
- Working with the Label Control
- Working with the Text Box Control
- Working with the Command Button Control
- Working with the Combo Box Control
- Working with the Frame Control
- Working with Option Button Controls
- Working with Control Appearance
- Setting the Tab Order
- Populating a Control
- Adding Code to Controls
- Launching a Form in Code

Working with the PivotTable Object
- Understanding PivotTables
- Creating a PivotTable Using Worksheet Data
- Working with the PivotTable Objects
- Working with the PivotFields Collection
- Assigning a Macro to the Quick Access Toolbar

Debugging Code
- Understanding Errors
- Using Debugging Tools
- Setting Breakpoints
- Stepping through Code
- Using Break Mode during Run mode
- Determining the Value of Expressions

Handling Errors
- Understanding Error Handling
- Understanding VBA's Error Trapping Options
- Trapping Errors with the On Error Statement
- Understanding the Err Object
- Writing an Error-Handling Routine
- Working with Inline Error Handling