

# Financial Analysis Systems Certificate

## *Using VBA for Excel*

### **Australia's premier Excel VBA course for finance professionals**



A 48-hour Excel short programming course for professionals in the finance & accounting professions looking for better automation & control over important financial information in an increasingly complex environment. VBA macro programming techniques are taught from the ground up.

#### **Modes of Study**

- In class mode: 48 hours over one month
- In residential mode: 56 hours over one week

#### **Key Outcomes**

- Learn a proven approach for developing automation in Excel-based projects;
- 48 to 56 hours of continuing professional education for accountants;
- Gain insights into the development of robust code for routine analytical & reporting tasks;
- Learn effective techniques for importing & validating data;
- Learn how to develop routines from scratch with intent, rather than relying on recorded code or kludging code discovered on the internet;
- Learn how to interact with the user through built-in & custom dialog boxes & VBA functions;
- Understanding of how to add your tools & macros to Excel's own menus or ribbon interface;
- Recognise when it is appropriate to use built-in features & when to add custom functionality.

#### **The Course Delivers**

- A thorough grounding in VBA language;
- Familiarity with the VBA integrated development environment;
- Familiarisation with the tools to work with & develop VBA code in Excel;
- An overview of the Excel object model, focusing on critical objects, properties & methods to develop effective, useful & productive code;
- Standardised layout & naming conventions are explained and demonstrated;
- Proven approaches to development of Excel VBA projects from planning to rollout;
- Effective design approaches for user interface & custom dialog boxes;
- Understanding of the debugging & testing procedures in the VBA development environment;
- Demonstrates how to construct custom functions to extend Excel & reduce complexity in workbooks;
- Practical guidance on the layout & structuring of workbooks for analysis & forecasting;
- Demonstrates how to trap and handle errors.

#### **Who Benefits**

- Financial modellers looking to automate key tasks in their models, such as Solver, or who wish to develop custom functions to manage spreadsheet complexity;
- Business Analysts responsible for routine forecasting & performance analysis who wish to automate and streamline reporting and analytical tasks, such as the importation of data, data cleansing;
- Fund accountants & managers in equity & property funds looking to automate reporting, valuation and report distribution tasks;
- Procurement, project planning & logistical staff looking to automate planning & reporting tasks;
- Anyone who uses Excel to perform their role, looking to automate routine tasks, such as data import, data cleaning, workbook distribution, data entry and reconciliation processes.

## Benefits for the Employer

- This course provides essential knowledge & design principals to aid staff in creating robust applications in Excel;
- It is designed to replace the commonplace ad hoc recording & copying of VBA code with code robustly designed for its purpose. Understanding of important issues such as the appropriate use of code & how to avoid common design flaws & inappropriate bypassing of important Excel controls & features is delivered.
- Many managers rightly have concerns about the unmanaged proliferation of macro code within the workbooks in their departments. If it is developed without clear guidelines & standards and is just kludged-together, then it represents a significant risk. However, well-designed code appropriately deployed can significantly improve productivity & can add to the control of errors in repetitive & time-sensitive tasks. Without training this objective is hard to achieve.
- The Financial Analysis Systems Certificate is specifically designed to take knowledgeable Excel users & not only impart essential technical knowledge to enable them to take the next step, but to canvass issues around design, anticipation & trapping of errors and to demonstrate proven techniques that quickly & robustly deliver solutions.
- Standardisation of approach, coding conventions and the re-use of code is an important outcome.

## About VBA & Macros

- VBA stands for Visual Basic for Applications. It is the automation language provided with Microsoft's Office suite of applications & enables the development of automation & the extension of the user interface in those applications.
- Macros are a common name for the procedures developed in this language. The actual derivation of the word goes back to the early days of PC computing. A macro (which should more properly be called a procedure) is a separate executable application or applet which can be attached to a tool bar, a button or other control, graphical objects or linked to an event in the host application to handle a procedural step, automate common actions, or respond to changes in host documents.
- Macros might be responsible for the automation of the importation of data from external sources, the changing of document contents, prompting a user to complete the entry of data to be stored in cells or forms.
- The language taught is VBA, not VB.NET. VBA and the integrated development environment provided as a standard extension of Office is the development language used & taught in the course. The .NET languages used in Visual Studio .NET can be used to develop add-ins and extensions for the Office application suite, but that is a more complex development environment than that used for this course.
- Note that VBA is not supported in Excel 2008 for the Macintosh. It was replaced by AppleScript in that version much to the annoyance of many users, although VBA has made a return in Excel 2011. It is only supported in Excel 2011 for the Macintosh.

## Is the content applicable to other Office applications?

- VBA is provided as a development language in most of Microsoft's Office suite. Most users, however, are generally interested in automating Excel, Access & Word. The course content relating directly to the VBA language, intrinsic data types, operators, dealing with the file system, calling user forms and the programming of common Office elements such as tool bars, context menus, the Ribbon and file search is applicable to all Office applications, so to that extent, therefore, the knowledge is transferrable.
- On the other hand, much of the course design rests on Excel specific features, such as worksheets, ranges, cells, formulas & Excel functions & is therefore heavily skewed to working with Excel & the Excel Object Model which exposes the elements of Excel & its features to the end-user programmer.
- The concepts taught about dealing with object models & the programmable elements of an application in a generalised way will apply to working with other object models, although the specifics will be very different. If you have specific questions relating to these matters, feel free to contact us to discuss them.
- The modules in the Financial Analysis Systems Certificate Extension Program do specifically cover working with Outlook, Word and PowerPoint from within Excel, and demonstrate techniques for automating those applications.

# Testimonials

Most of these testimonials were offered some months after course completion, not just as polite feedback on the final day of the course.

- “I just wanted to say thank you. The skills you taught and knowledge you imparted during the course I attended have made a real difference in my work. I am enjoying my days immensely and doing some interesting and innovative work, no longer limited - at least, not so much - by my (in)ability with Excel.” - Todd A
- “The course was fantastic. It was full of useful and practical methods that can be applied to my daily tasks.” - Erica H
- "I would like to say thanks for the inspirational course! I've found that my productivity at work has gone through the roof. Without you, none of that would have been possible. So thanks again. " - Simon S
- “Great examples used, great responses to real life situations, great presenter” - Fernando V
- “Your passion and patience was really, really appreciated. I was a bit worried when the course started that it would be too much for me, but following your guidelines, I have surprised myself just how much I can achieve. This has been a great add-on to my professional skill set! Thanks so much!” - Gemma S

## Detailed Course Outline

### Introductory macro techniques using VBA

- Introduction to macros & VBA
- Macro recording & editing recorded macros
- How to record generalised macros
- Recording macros with absolute references
- Recording macros with relative references
- Where to store macros
- Locating & activating macros
- Assigning recorded macros to worksheet controls

### Introduction to the IDE

- introduction to elements of VBA
- VBA keywords & syntax
- Operators & Statements
- Procedure naming & declarations
- Intrinsic data types
- Variable naming conventions & declarations
- Object variables
- Initialising & referencing objects
- Array variables, declaration & re-dimensioning
- Function Arguments
- Lifetime, visibility & scope of variables & constants
- Use of constants to aid maintenance
- Naming conventions
- The Excel Object Model
- The Office Object Model
- The MS Forms 2.0 Object Model
- Module & procedure naming
- Compiler options: Option Explicit, Option Base
- Compilation & compile on demand

### Program flow control

- If statements
- If blocks
- Select Case blocks
- Do & While blocks
- For loops using counters
- For Each loops using object variables
- With blocks

### Layout & Programming Style

- Referencing conventions
- Naming conventions for procedures, variables & constants
- Use of With statements to improve layout
- Code layout conventions & indentation to create more readable code blocks to aid in maintenance & debugging
- Use of subroutine calls
- Use of function calls
- common syntax errors & their correction

### Procedure & Function declaration

- When to use modules & worksheet modules
- Procedure declaration & scope
- Declaring arguments
- Determining argument & return value types
- Creating optional arguments & default values
- Passing values to subroutines vs global variables
- Passing object references
- ByVal & ByRef keywords
- Calling subroutines & functions
- Handling function return values

## Working with the Workbooks collection

- Working with Application object (Excel)
- Working with the Workbooks collection
- Working with the Workbook object
- Key workbook properties
- Creating new workbooks
- Creating new workbooks from templates
- Opening, closing & saving workbooks
- Responding to key workbook events
- Changing key workbook properties

## Working with the Sheets collections

- The Sheets, Worksheets & Charts collections
- The Worksheet, Chart & DialogSheet objects
- Key sheet properties
- Selecting & activating sheets
- Creating new sheets
- Creating new sheets from templates
- Adding, deleting, copying & moving sheets
- Responding to key worksheet events
- Changing key sheet properties
- Page setup & printing sheets
- Sheet visibility & sheet protection
- Visibility & protection effects on programming
- Iterating through workbooks with mixed sheet types

## Working with Ranges

- The Range object
- Cells collection
- Selecting or referencing a cell or cells
- Selecting or referencing a row or rows
- Selecting or referencing a column or columns
- Selections using SpecialCells & CurrentRegion
- Navigating through the cells collection of a range
- Working with the ActiveCell property
- Working with the Selection property
- Reading & changing the values of key properties, such as Formula, FormulaR1C1, Value & Text
- Using Find to locate matching cell or cells
- Working with named ranges
- Selecting & updating ranges
- Selecting & updating dynamic ranges
- Selecting ranges by address or by Name reference
- Using Styles & formatting properties to change the appearance of ranges
- Updating formulas & values
- Working with Copy, Paste & PasteSpecial
- Replacing Copy & Paste with direct updates

## Working with the Names collection

- Reducing code load, allowing for user interaction & maintenance through the use of Names
- Key application settings effectively managed through Names
- Working with the Names collection
- Working with the Name object
- Key Name properties
- Creating, deleting & editing Names
- Selecting named ranges
- Retrieving values from & updating values of named ranges

## Working with the Styles collection

- Reducing code load allowing for user interaction & maintenance through the use of Styles
- Working with the Styles collection
- Working with the Style object
- Key style properties
- Creating, deleting, editing & applying Styles
- Removing unwanted Styles from a workbook

## Custom functions

- Creating custom functions to manage formula complexity
- Creating custom functions to provide specialised calculations not provided by the Excel function library
- Rules for functions to be called from a worksheet
- Return values & arrays
- Handling argument errors & returning errors from functions
- Detecting missing arguments & argument types
- Default values for arguments
- Sample custom functions
- Wrapping built-in functions to add value
- Deploying custom worksheet functions

## Advanced techniques to improve performance

- Locking screen updates
- Selecting appropriate variable types
- More efficient methods of referencing
- Bypassing the clipboard
- Indicating progress to the end user
- Locating a specific workbook in memory
- Prompting for the selection of a workbook from a directory

## MS Forms 2.0 & custom dialog boxes

- Understanding control types
- Understanding control properties
- Understanding control events & event execution order
- Understanding containers & the controls collections
- Responding to control events
- Selecting appropriate controls
- Setting default values
- Responding to user inputs & selections
- Validating user inputs & selections
- Interpreting & formatting text box values, especially problematical values
- Dynamically linked controls
- Designing custom dialog boxes
- Instantiating dialog boxes
- Dialog box life cycle
- Referencing controls in dialogs
- Transferring data between the worksheet, named ranges & dialog box controls
- Determining which button closed a dialog box
- Tab order, accelerators & short cuts

## Error handling

- Error object & key properties
- On Error statement
- Writing standard error handlers
- Trapping & responding to errors
- Ensuring clean-up code executes
- Closing opened workbooks, without saving changes
- Resetting the user interface, undoing changes

## Sample Project - Data Bank for Business Reporting

- The Data Bank Reporting Model
- Menus & sub-menus to provide access to application features
- Customising the Fluent UI Ribbon using XML & the Custom UI editor
- Workbook event handling macros
- Storing user selections & session variables
- Sharing settings with worksheet functions via named ranges
- Techniques for assisting the end-user with on-line information: formatted help screens & hyperlinked documents
- Data Bank reporting: extracting data from the Data Bank, print macros & page set-up
- Reporting using file linking & reporting using copied data
- Simple dialog boxes
- Sample documentation

## Sample Project – Lease Amortisation Schedule

- Utilises a custom dialog box to prompt the user for information regarding the lease
- Loads data from worksheet to the dialog box
- Interacts with the user & responds to user input
- Stores the user entries in the worksheet & updates the report
- Complex dialog boxes
- A demonstration of the start-to-finish process of using dialog boxes in Excel applications

## Key Excel techniques

- Working with the Application object to manage core settings & access key references
- Working with the Workbooks collection & referencing workbooks by code
- Working with worksheets & the Worksheets, Charts & Sheets collections
- Working with range objects & understanding the Range, Cells, Rows, Columns, Offset, Resize & Areas properties
- Range validation
- Working with ranges using names
- Range formatting, using Styles, Font, Interior, Borders properties & their associated object structures
- Entering formulas under macro control
- Accessing & updating the values in cells using Value, Value2 & Text properties
- Using counted For loops & For Each loops to process blocks of cells
- Copying & pasting data using the clipboard
- Calling built-in Excel dialog boxes
- Utilising Excel's built-in functions to optimise code

## Testing & Debugging VBA Code

- Program testing & debugging
- Debugging with step mode
- Using break points
- Using watch window & locals window to view variables
- Using the Immediate Pane to evaluate expressions
- Interpreting run-time errors
- Trapping & responding to run-time errors
- Planning for recovery



## Workshop Exercises

- A set of review exercises are worked at regular intervals throughout the course that require the students to work in groups to implement the knowledge gained in solving the problems set.
- Commentaries and sample solutions are provided to the students for the review exercises. The results of these exercises are critiqued in class & alternative approaches to solving these problems are canvassed.
- Final assessment (optional for in-house courses) is based upon a longer workshop during which the students working in groups to demonstrate their knowledge & analysis skills through solving a set of problems based around the materials covered. Note: Sample solutions and commentaries are not provided for the assessed questions.

## Public Courses

Public offerings of the Financial Analysis Systems Certificate are available in Sydney, Melbourne, Brisbane, Adelaide & Perth. For details, visit our web site ([www.clarksonitt.com/calendar](http://www.clarksonitt.com/calendar)) or call +61 (0) 417-777-750.

## Residential Mode Courses

We run some of the public offerings in residential mode. These more intensive sessions are suitable for those who would like to focus on the training & complete it in a concentrated period of study. It also makes it easier for remote students to attend a course that is completed over one week, reducing travel time and costs. Day-only students are welcome at all residential mode courses. They also offer opportunities for extra one-on-one coaching.

## Private Courses

In-house offerings of the Financial Analysis Systems Certificate can be arranged anywhere in Australia & we can assist with the hire of suitable training facilities where not available in-house. Requirements for facilities are discussed on our web site. For a quotation or to discuss your particular needs, call +61 (0) 417-777-750.

## About the Course Author and Lead Presenter



Following an early career in both public and commercial accounting, James Clarkson has more than twenty-five years' experience as a presenter of professional programs and more than thirty consulting & model building for corporate clients.

He has gained wide experience in model & analytical report design & development and also has wide experience in application development using Microsoft Office (especially Excel), Visual Basic & Visual Basic for Applications (VBA).

He is an outstanding educator, demonstrating the ability to clearly explain difficult and technical subjects and consistently rates above 90% for satisfaction across a number of measures on student evaluations.

## Excel Version

- The course materials are fully compatible with all versions of Excel from Excel 2007 to Excel 2013. Version specific instructions are given where these vary between versions.
- Public courses are conducted using Excel 2010
- Note that for the Macintosh, VBA is not supported in Excel 2008, only Excel 2011 for the Macintosh may be used.

## Course Materials

- The course materials include a 350+ page course text & a 24 page ready reference
- A full set of blank and worked sample workbooks as well as a number of reference documents are available for download online.