PREMIER

Microsoft Excel 2007 Advanced

Premier Training Limited

4 Ravey Street

London

EC2A 4QP

Telephone +44 (0)20 7729 1811

www.premcs.com

TABLE OF CONTENTS

INTRODUCTION

This manual is designed to provide information required when using *Excel 2007*. This documentation acts as a reference guide to the course and does not replace the documentation provided with the software.

The documentation is split up into modules. Within each module is an exercise and pages for notes. There is a reference index at the back to help you to refer to subjects as required.

These notes are to be used during the training course and in conjunction with the *Excel 2007* reference manual. *Premier Computer Solutions* holds the copyright to this documentation. Under the copyright laws, the documentation may not be copied, photocopied, reproduced or translated, or reduced to any electronic medium or machine readable form, in whole or in part, unless the prior consent of *Premier Computer Solutions* is obtained.

MODULE 1
REVIEW OF INTERMEDIATE COURSE

Revision Exercise

1. Create the exercise on the following page, using the calculations given.

2. The factors that might change are located in separate cell for easy "what-if" analysis. The formulae are given below:

SALES start at 3,500 and increase by the Assumption Growth in Sales "10%".

PRICE starts at 15.50 and increases by the Assumption Growth in Price "0"

REVENUE is Sales * Price

RAW MATERIALS are Sales * Assumption Unit Raw Material

LABOUR is constant at 10,000

ENERGY is Sales * Assumption Unit Energy

DEPRECIATION is constant at 750

TOTAL COSTS is the sum of the above four costs

GROSS PROFIT is Revenue – Total Costs

OVERHEADS are constant at 12,500

NET PROFIT is Gross Profit – Overheads

The Net Profit for December should be £60,631.63

3. Rename the worksheet as **PROFIT PROJECTION**.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	**Profit Protection - Widget Division 2008**													
2														
3		Jan-09	Feb-09	Mar-09	Apr-09	May-09	Jun-09	Jul-09	Aug-09	Sep-09	Oct-09	Nov-09	Dec-09	
4														
5	Sales	3500												
6	Price	15.5												
7	**Revenue**													
8														
9	Raw Materials													
10	Labour	10000												
11	Energy													
12	Depreciation	750												
13	**Total Costs**													
14														
15	Gross Profit													
16	Overheads	12500												
17														
18	**Net Profit**													
19														
20	**Assumptions**													
21	Growth in Sales as %	10%												
22	Growth in Price as %	0%												
23	Unit Raw Material	6.75												
24	Unit Energy	0.35												
25														

MODULE 2
NAMING RANGES

There are a variety of uses for names in a workbook. A name can be applied to any cell or range. Names are also useful for the following:

- ❑ Making formulas easier to understand
- ❑ Quick Navigation
- ❑ Improving Solver's report results
- ❑ Storing a value that will be used over and over but that might occasionally need to change, such as a sales tax rate.
- ❑ Storing formulas
- ❑ Defining a dynamic range

NAMING A CELL, RANGE OR FORMULA

The following must rules must be followed when naming ranges:

- ❑ The first character if a range name must be a letter or underline.
- ❑ The remaining characters must be letters, numbers, underlines or periods.
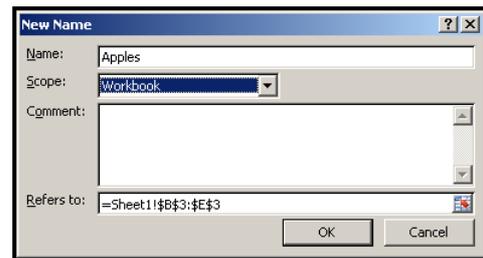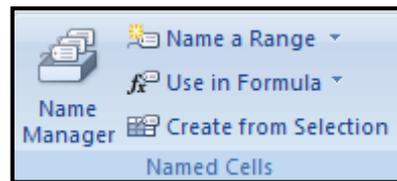- ❑ No spaces.
- ❑ Do not use cell references as names.

1. Select the cell(s).

2. Click in the Name Box in the formula bar.
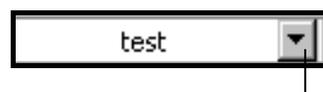
Name Box

3. Type a name (named ranges cannot contain any spaces) and then press **ENTER**.

Alternatively, on the Formulas tab, in the Named Cells group, click Name a Range.

1. In the **New Name** dialog box, in the **Name** box, type the name that you want to use for your reference. Names can be up to 255 characters in length.

2. To specify the scope of the name, in the Scope drop-down list box, select Workbook, or the name of a worksheet in the workbook.

3. Click on **OK**.

Create Names Based On Row/Column Titles

1. Select the range you want to name, including the row or column titles you want to use for the names.

2. Click on the **Formulas** tab, in the **Named Cells** group, click **Create from Selection**.
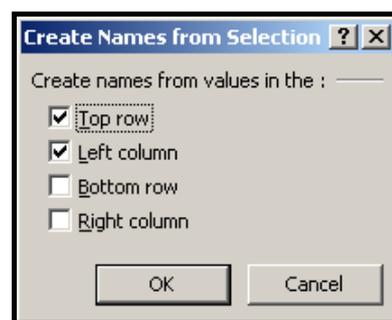
3. Select the appropriate check box or boxes to name the rows or columns using the text in the top row, bottom row, left column, or right column of the range.

4. Click on **OK**.

Using Named Ranges

Named Ranges can be used to move to various locations in a

workbook and pasted into formulas.

MOVING TO A NAMED RANGE

1. Click on the downward arrow to the right of the name box in the formula bar.

2. Select the named range.

Name Box List

PASTING NAMES IN FORMULAS

1. Start formulas by typing = and then the formula name.

2. Click on the **Formulas** tab, in the **Named Cells** group, click **Use in Formula**.
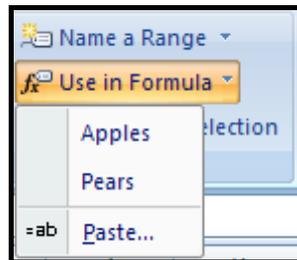
3. Select the name to use.

4. Finish the formula and then press **ENTER**.

5. Alternatively press **F3** to display the **Paste Name** dialog box.
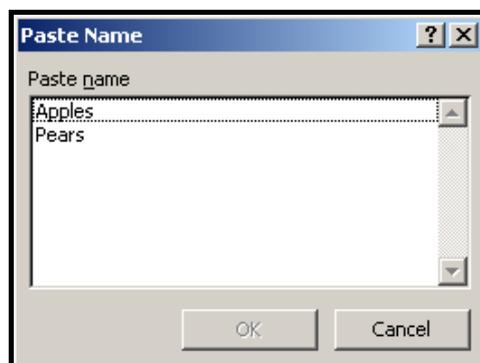
DELETING A NAMED RANGE

1. Click on the **Formulas** tab, in the **Named Cells** group, click **Name Manager**.
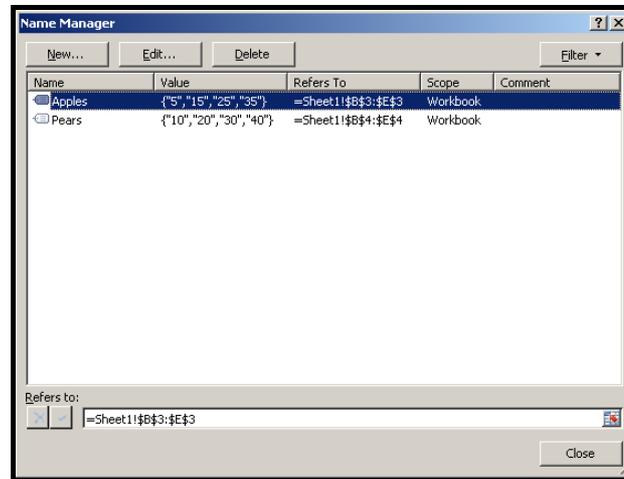
2. Select the name to delete and click on the **Delete** button.

3. Click on **OK**.

4. Alternatively to display the **Name Manager** dialog box press **CTRL + F3**.

PASTE A LIST OF NAMED RANGES

1. Select an empty cell.

2. Click on the **Formulas** tab, in the **Named Cells** group, click **Use in Formula**.

3. Choose **Paste**.

4. Click on the **Paste List** button.
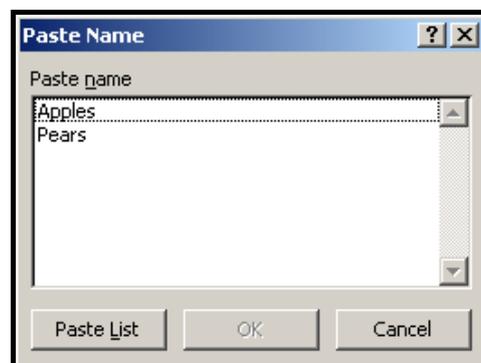
5. Alternatively press **F3** or

test	=Sheet1!$C$4,Sheet1!$C$6:$C$11,Sheet1!$E$6:$E$10
Test1	=Sheet1!$C$5:$C$9
Test2	=Sheet2!$F$7:$F$12

Notes – Module 2

Notes – Module 2

Notes – Module 2

Notes – Module 2

MODULE 3
FUNCTIONS

Functions are built-in formulas that perform complex mathematical, financial, statistical or analytical calculations. Excel provides more than 200 built-in functions, or predefined formulas.

Each function consists of an equal sign (=), the function name and the argument(s). Arguments are cells used for carrying out the calculation. The SUM function adds the number of cells in specified cells.   The active cell shows the result of the function.

$$=SUM(B1:B9)$$

| Equal | Function Name | Arguments |
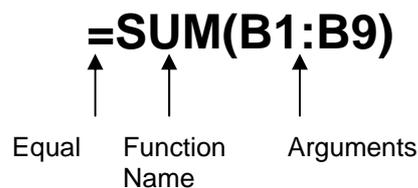
IF STATEMENTS

A logical function enables you to make a decision depending if the conditions set are true or false.

IF FUNCTION

=IF(logical_test,value_if_true,value_if_false)

An example is in cell A3 if the value is equal to 10 insert 1, if not insert 0.

	A	B	C
1			
2			
3	10	1	

In cell B3 the calculation would be

=IF(A3=10,1,0)

If the true or false condition is to be text and not a value, the text has to be enclosed in double quotes.

=IF(A3=10,"Yes", "No")

Listed below are the comparative operators that can be used in logical functions.

=	Equal To
<	Less Than
>	Greater Than
<=	Less than or equal to
>=	Greater than or equal to
<>	Not Equal to

NESTING =IF()'s

You may want to use an =IF() function again as part of the TRUE or FALSE part of the formula.

If Average Score is	Then return
Greater than 89	A
From 80 to 89	B
From 70 to 79	C
From 60 to 69	D
Less than 60	F

You can use the following nested =IF() function:

*Note: Remember to close the same number or brackets at the end of the function as functions, there are four **=IF()** to four brackets at the end. There can only be a maximum of **7** nested =IF()'s in a single function.*

=IF(AverageScore>89,"A",IF(AverageScore>79,"B",IF(AverageScore>69,"C",IF(AverageScore>59,"D","F"))))

The function on the previous page reads:

If the average score is greater than 89 then insert an A, if not is it greater than 79, if it is insert a B, if it is not, then is it greater than 69, if it is insert a C if not is it greater than 59, if it is insert an D if not then insert F.

Using Other Functions with IFs

For example if the =SUM() function is nested within an =IF() function, the condition of the =IF() function can be based on whether the total value in a range of cells is above or below a certain value.

	A	B	C
1	Sales figures for Qrt1 1997		
2			
3	Date	Total Sales	
4	01-Jan	30000	
5	01-Feb	25689	
6	01-Mar	45689	
7		Good Sales	
8			
9			

The following function would be used in cell B7.

=IF(SUM(B4:B6)>9000, "Good Sales", "Bad Sales")

=AND(logical1, logical2)

This logical function compares whether two cell match the condition required, if the cells meet the condition TRUE is inserted into the cell if not then FALSE is used.

	A	B
1		
2	10	
3	20	
4	30	
5		TRUE
6		

The function in cell B5 is:

=AND(A3=20,A4=30)

=OR(logical1, logical2)

This logical function compares whether one of two cells meets the condition required. If the cells meet the condition TRUE is inserted into the cell if not then FALSE is used.

	A	B
1		
2	10	
3	30	
4	40	
5		FALSE

The function in cell B5 is:

=OR(A3=20,A4=30)

Using The =AND() and =OR() Function With =IF()s

	A	B
1		
2	10	
3	30	
4	40	
5		This is Wrong

The function in cell B5 is:

=IF(OR(A3=20,A4=40), "This is Correct", "This is Wrong")

or using the =AND() function:

=IF(AND(A3=20,A4=40), "This is Correct", "This is Wrong")

TEXT FUNCTIONS

=TEXT(Value,Format_Text)

This function converts a number to a formatted text entry.

> **Value**   is the number or cell reference to a cell containing a value.

> **Format_Text**   is the format from the category box in the Format Cells/Number dialog box. This cannot be the General format.

   =TEXT(1.5, £0.00)   the result would be £1.50

=VALUE(text)

This function converts a text string that represents a number to a number.

Text   can be text enclosed in quotation marks or a cell reference.

=VALUE("£1,000") equals 1,000

=LEFT(text,num_char)

This function returns the first leftmost characters in a string.

=RIGHT(text,num_char)

This function returns the first characters in a string from the right.

Text   can be text enclosed in quotation marks or a cell reference.

Num_char represents the number of characters from the left/right that you want the function to return.

	A	B	C
1			
2	West Region	West	Region
3	North Region	Nort	Region
4	South Region	Sout	Region
5	East Region	East	Region
6			

Note: These functions count the spaces and punctuation as characters and is only effective if all pieces of text are the same length.

In cell B2 the following formula was inserted

=LEFT(A2,4)

In cell C2 the following formula was inserted

=RIGHT(A2,6)

LEN(text)

This functions counts the number of characters in a string, this includes spaces and pronunciation.

Text   can be text enclosed in quotation marks or a cell reference.

	A	B
1		
2	West Region	11
3		

In cell B2 the following formula has been used:

=LEN(A2)

=CONCATENATE(text1,text2…)

This function allows you to join different text cells together to produce a single string of text.

	A	B	C
1	**First Name**	**Surname**	
2	John	Black	John Black
3	Peter	Kelly	Peter Kelly
4	Paul	Green	Paul Green
5	Simon	Smith	Simon Smith
6	Andrew	Guy	Andrew Guy
7			

In column C the first and surnames have been joined together.

=CONCATENATE(A2, " ", B2)

To insert a space between each entry place a space between two quotes.

Alternatively you can use the ampersand (&) to join text cells together.

=A2&" "&B2

DATE AND TIME

=TODAY()

This function will return the current date, it does not require any arguments and will update automatically.

=NOW()

This function will return the current date and time, it does not require any arguments and will update automatically.

=MONTH(serial_number)

This function returns the number of the month (from 1 – 12) corresponding to a serial number.

	A	B
1		
2		
3	23/05/98	5
4		

In cell B3 the following formula was entered:

=MONTH(A3)

LOOK UP FUNCTIONS

The LOOKUP functions are a good method of looking for information in a list. It searches the left most column of a list for a particular value and returns an entry in a column to the right.   There are two LOOKUP functions one that lookups vertically, VLOOKUP or horizontally, HLOOKUP.

=VLOOKUP(lookup_value,table_array,col_indx,range_lookup)

Lookup_value	is the value that you require the function to lookup.
Table_array	is the range of cells that you want to look in. You can name cell rangers as opposed to using cell references
Col_indx	this is the column in the table_array from which the matching value should be returned. If you enter 2 it will return the value in the second column, 3 would return a value in the third column and so on.
Range_lookup	this is a logical test. If you want the LOOKUP function to insert TRUE or FALSE when a match is made enter either TRUE or FALSE. When omitted the actual entry is returned.

Note: The table must be sorted in ascending order. The VLOOKUP function only returns one answer. If there are several occurrences of a target value VLOOKUP stops at the first one. It is not used for entries that appear more than one in a list.

	A	B	C	D	E	F	G
1	Accounts Payable						
2							
3	Number	Date Rec	Date Due	Amount		Enter Invoice Number	3496
4	3484	04/12/97	01/12/97	325.56			
5	3487	09/12/97	10/12/97	123.45		**Amount**	654.78
6	3488	04/01/98	10/01/98	321.65			
7	3492	05/01/98	12/12/97	456.98			
8	3496	12/01/98	15/01/98	654.78			
9	3497	13/01/98	15/01/98	741.25			
10	3498	21/01/98	15/01/98	963.85			
11	3504	22/01/98	10/01/98	951.35			
12							

In the above example the invoice number is entered in cell G3

and the amount in G5 will be looked up on the table.

Cell G5 contains the following:

=LOOKUP(G3,A3:D11,4)

=HLOOKUP((lookup_value,table_array,row_indx, range_lookup)

Looks for a value in the top row of a table and returns the value in the same column for the row you specify.

Lookup_value	is the value that you require the function to lookup.
Table_array	is the range of cells that you want to look in. You can name cell rangers as opposed to using cell references.
Row_index	this is the row in the table_array from which the matching value should be returned. If you enter 2 it will return the value in the second row, 3 would return a value in the third row and so on. Headings in a table count as row 1.
Range_lookup	this is a logical test. If you want the LOOKUP function to insert TRUE or FALSE when a match is made enter either TRUE or FALSE. When omitted the actual entry is returned.

	A	B	C	D	E	F
1	Using HLOOKUP to lookup value by date and not store					
2						
3	Date	05/03/97				
4	Store	2				
5						
6	Result	£987.00				
7						
8	**Store**	**05/01/97**	**05/02/97**	**05/03/97**	**05/04/97**	
9	Store 1	£147.00	£852.00	£789.00	£159.00	
10	Store 2	£258.00	£741.00	£987.00	£357.00	
11	Store 3	£369.00	£123.00	£456.00	£842.00	
12	Store 4	£963.00	£654.00	£321.00	£862.00	
13						

Cell B6 contains the following:

=HLOOKUP(B3,A8:E12,3)

FINANCIAL FUNCTIONS

=PMT(rate,nper,pv,fv,type)

This function calculates the payments on a loan (principle) at a given interest rate for a specified number of payment periods (term). It is normally used to calculate the monthly repayments of a mortgage that is based on a constant interest rate.

Rate	is the interest rate. If you are calculating monthly mortgage payments for a loan with an annual interest of 8%, it has to be divided by 12 to get the monthly interest rate.
Nper	is the total number of months for the repayments. This has to be shown in months.
Pv	is the present value of the loan
Fv	is the future value, or a cash balance you want to attain after the last payment is made. If fv is omitted, it is assumed to be 0 (zero), that is, the future value of a loan is 0. This argument is optional
Type	is the number 0 (zero) or 1 and indicates when payments are due. This argument is optional
	0 or omitted indicates that payments are due at the end of the period
	1 indicates that payments are at the beginning of the period.

Cell B7 contains the following formula

=PMT(B4/12,B5*12,B3)

=PPMT(rate,per,nper,pv,fv,type)

This function returns the principle for an investment based upon periodic, constant payments and a constant interest rate.

=IPMT(rate,per,nper,pv,fv,type)

This function returns the interest paid for an investment based upon periodic, constant payments and constant interest rate.

Rate	is the interest rate. If you are calculating monthly mortgage payments for a loan with an annual interest of 8%, it has to be divided by 12 to get the monthly interest rate.
Per	Specifies the period and must be in the range of 1 and the nper..
Nper	is the total number of months for the repayments. This has to be shown in months.
Pv	is the present value of the loan
Fv	is the future value, or a cash balance you want to attain after the last payment is made. If fv is omitted, it is assumed to be 0 (zero), that is, the future value of a loan is 0. This argument is optional
Type	is the number 0 (zero) or 1 and indicates

when payments are due. This argument is optional

0 or omitted indicates that payments are due at the end of the period

1 indicates that payments are at the beginning of the period

	A	B	C
1	**Calculating Mortgage Payments**		
2			
3	Loan	£100,000	
4	Interest Rate per annum	8%	
5	Number of Years	25	
6	Current Period (1st Payment)	1	
7			
8	Payment (PMT)	-£771.82	
9			
10	Principle on Payment (PPMT)	-£105.15	
11	Interest on Payment (IPMT)	-£666.67	
12			

Cell B10 contains the following function:

=PPMT(B4/12,B6,B5*12,B3)

Cell B11 contains the following function

=IPMT(B4/12,B6,B5*12,B3)

=PV(rate,nper,pmt,fv,type)

This function calculates the present value of an investment based on the total amount that a series of future payments is worth now.

=NVP(rate,value1,value2)

This function calculates the net present value of an investment by using a discount rate and a series of future payments (negative values) and income (positive values).

	Rate	Is the rate of discount over the length of one period.
	Value1, value2,	are 1 to 29 arguments representing the payments and income.

	A	B	C	D
1	The Present Value of Unequal Regular Payments			
2				
3	Period	Payment		
4	1	2000		
5	2	2000		
6	3	2000		
7	4	2500		
8	5	3000		
9	6	3000		
10	7	3500		
11				
12	Discount Rate	6%		
13	Net Present Value	£14,011		
14				

Cell B13 contains the following function

=NPV(B12,B4:B10)

=FV(rate,nper,pmt,pv,type)

If you plan to save a regular amount at regular intervals, the =FV function calculates how much you will have saved at a specific point in the future, give a constant compounding rate.

	Rate	Is the periodic compounding rate on your investment
	Nper	How many periods in the saving plan.
	Pmt	How much you are paying each period.
	Pv	This is an optional argument that lets you specify a lump sum already present at the beginning of the saving plan.
	Type	Is the number 0 (zero) or 1 and indicates when payments are due. This argument is optional
		0 or omitted indicates that payments are due at the end of the period

1 indicates that payments are at the beginning of the period.

For example, if you plan to deposit £2,000 each year for the next 20 years into an account to save for retirement. The account pays 7.5% interest, compounded annually, interest is paid on the last day of each year. To calculate the value of your account in 20 years.

	A	B	C
1	The Future Value of Regular Equal Payments		
2			
3	Payment	-£2,000	
4	Periodic Compounding Interest	7.5%	
5	Number of years	20	
6			
7	Future Value	£86,609.36	
8			

Cell B7 contains the following formula:

=FV(B4,B5,B3)

MATHEMATICAL FUNCTIONS

=ROUND(number,num_digits)

This function rounds number to a specified number of decimal places.

=ROUNDDOWN(number,num_digits)

This function rounds down numbers towards zero.

=ROUNDUP(number,num_digits)

This function rounds up numbers away from zero.

Function	Result
=ROUND(333.1451,2)	333.15
=ROUND(333.1451,-2)	300
=ROUNDDOWN(333.1451,2)	333.14
=ROUNDUP(333.1451,1)	333.2

=ABS(number)

This function returns the absolute value of a number. A number without a sign for example a negative value will become positive.

Function	Result
=ABS(-500)	500

=SUMIF(range,criteria,sum_range)

This function adds cells specified by a criteria.

=COUNTIF(range,criteria)

This function counts the number of cells specified by a criteria.

	A	B	C	D
1	SUMIF and COUNTIF Functions			
2				
3	Name	Sales		
4	Anderson	150	Total Sales for Green	1687
5	Green	256		
6	Harry	236	How many times does Harry appear?	3
7	Green	456		
8	Harry	789		
9	Anderson	987		
10	Green	654		
11	Green	321		
12	Harry	147		
13	Anderson	258		

Cell D4 contains the following:

=SUMIF(A4:A13, "Green",B4:B13)

Cell D6 contains the following:

=COUNTIF(A4:A13, "Harry")

	A	B	C
1	**The Present Value of Regular Payments**		
2			
3	Payments per month	£200	
4	Periodic Interest Rate per annum	7.5%	
5	Number of Years	10	
6			
7	Present Value	-£16,849	
8			
9			

INFORMATION FUNCTIONS

=ISBLANK(value)

This function returns TRUE or FALSE if the cell it is referenced to is blank.

=ISBLANK(A10)

If A10 was empty the result would be FALSE.

=ISNA(value)

This function return TRUE or FALSE if the cell it is referenced to returns the #N/A error message.

=ISNA(A10)

If A10 had the error #N/A the result would be TRUE.

=ISERROR(value)

This function return TRUE or FALSE if the cell it is referenced to returns one of the following error messages.

#N/A, #VALUE, #REF, #DIV/0, =NUM, #NAME?, #NULL

=ISERROR(A10)

If A10 displayed an error the result would be TRUE.

=ISERR(value)

This function return TRUE or FALSE if the cell it is referenced to returns one of the following error messages.

#VALUE, #REF, #DIV/0, =NUM, #NAME?, #NULL

=ISERR(A10)

If A10 displayed an error (not #N/A) the result would be TRUE.

Used with the =IF() function the =IS() functions can highlight to users errors in the worksheet

For example stopping the cell returning #DIV/0

	A	B	C
1	**ISERR**		
2			
3	10		
4			
5	Enter a number in to cell A4		
6			
7			

In A5 the cell would show #DIV/0 when cell A3 is divided by A4. The following calculation would be typed in A5 to display a message to insert a value in cell A4.

Note: =ISERROR() can also be used

=IF(ISERR(A3/A4)=True, "Enter a number in to cell A4", A3/A4)

SUBTOTALS

Excel enables you to summarise sorted data in a list with subtotals. When a list is summarised with subtotals, *Excel* calculates the subtotals based on subsets of the data and also calculates a grand total.

CREATING SUBTOTALS

1. Sort the data according to the order in which you want to create subtotals. For example to generate subtotals for each region, sort the list in to regions.

	A	B	C	D	E	F	G
1	**SubTotals**						
2							
3		**Product**	**Store**	**Region**	**Month**	**Amount**	
4		CBT	Norwich	East	Mar	987	
5		Book	Norwich	East	Jan	357	
6		Video	Glasgow	Scotland	Feb	456	
7		Book	Glasgow	Scotland	Apr	654	
8		Book	London	South	Jan	123	
9		Video	London	South	Mar	321	
10		CBT	London	South	Mar	753	
11		Video	Bristol	West	Jan	789	
12		Book	Bristol	West	Feb	159	
13							

2. Select a cell in the list you want to summarise.

    For example any cell in column F.

3. Click on the Data Ribbon and select Subtotals.
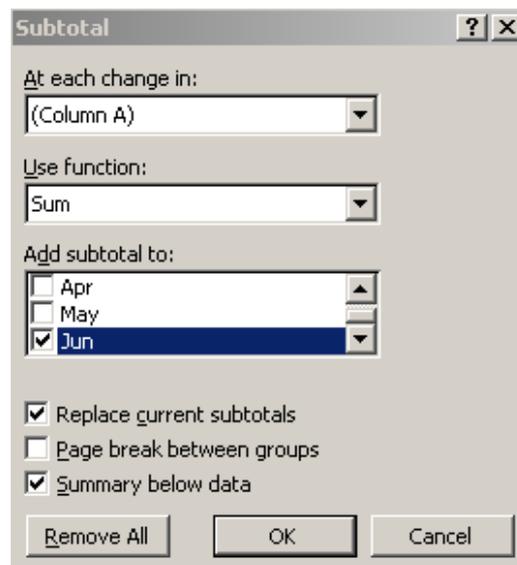
4. In the "At each change in" drop-down list select in this case region. This will generate a subtotal at each change in region.

5. Select a function from the "Use function" drop-down list.

6. Select the data you want to subtotal in the "Add subtotal to:", in this example select amount.

7. Click on OK.

1 2 3		A	B	C	D	E	F
	1	Product Sales with subtotals					
	3		Product	Store	Region	Month	Amount
	4		CBT	Norwich	East	Mar	987
	5		Book	Norwich	East	Jan	357
	6				East Total		1344
	7		Video	Glasgow	Scotland	Feb	456
	8		Book	Glasgow	Scotland	Apr	654
	9				Scotland Total		1110
	10		Book	London	South	Jan	123
	11		Video	London	South	Mar	321
	12		CBT	London	South	Mar	753
	13				South Total		1197
	14		Video	Bristol	West	Jan	789
	15		Book	Bristol	West	Feb	159
	16				West Total		948
	17				Grand Total		4599

Above is an example of the total amount sold for each region.

HIDING AND DISPLAYING DATA

When you add automatic subtotals to a list, *Excel* displays the list in Outline view. You can expand and contract the level of detail in the list to display only the subtotals and grand totals of data.

1. To the left of the worksheet there are three buttons labelled 1,2, & 3.

2. To view the sub and grand totals click on the button labelled 2.

3. To view just the grand total click on the button labelled 1.

4. To view all in the list click on the button labelled 3.

1 2 3		B	C	D	E	F	
	1	ales with subtotals					
	2						
	3	Product	Store	Region	Month	Amount	
	6			East Total		1344	
	9			Scotland Total		1110	
	13			South Total		1197	
	16			West Total		948	
	17			Grand Total		4599	
	18						
	19						

Level 1, 2 and 3 buttons

Show detail level button

Hide Detail Level button

5. To display the detail for a level click on the plus (+) button.

REMOVING SUBTOTALS

1. Click on the Data Ribbon and select Subtotals.

2. Click on the "Remove All" button.

CREATE "NESTED," OR MULTIPLE-LEVEL, SUBTOTALS

To "nest," or insert, subtotals for smaller groups within existing subtotal groups, you must first sort the list.

1. Sort the list by two or more columns for which you want to calculate subtotals. For example, to summarise amount sold by region and by store within each region, first sort

the list by the region column, and then sort by the store column. When *Excel* inserts subtotals, the stores subtotal values are nested within the subtotal values for the region.

2. Insert automatic subtotals for the first column for which you want subtotals. This column should be the column you specified in the "Sort by" box when the list was sorted. (In the example, the region column would be the first column.)

3. After you have displayed automatic subtotals for the first column, repeat the process for the next column.

4. Select a cell in the list.

5. Click on the Data Ribbon and select Subtotals.

6. In the "At each change in" box, select the next column you want to subtotal.

7. Clear the "Replace current subtotals" check box, and then click on OK.

Note: Subtotal summaries can be used to create charts.

8. Repeat steps 4 through 7 for each column for which you want to calculate subtotals.

EXERCISE - MODULE 3

	A	B	C	D	E	F	G
1				Tax Analysis			
2							
3	Drink	Tax Band	Price	Sales	Total	Tax	
4	Whisky	B	12.90	8,000			
5	Brandy	C	14.50	6,890			
6	Gin	A	8.65	12,896			
7	Vodka	D	9.95	12,356			
8	Rum	B	11.30	7,896			
9	Port	D	13.90	2,300			
10							
11				Totals			
12							
13							
14	Tax Band	Rate		RATE A			
15	A	0%		RATE B			
16	B	5%		RATE C			
17	C	8%		RATE D			
18	D	10%					
19							
20							

1. Create the above worksheet.

2. Enter a formula to calculate the TAX at 10%, but only if the TOTAL exceeds 90,000. If the TOTAL does not exceed 90,000 insert 0.

3. Replace the previous formula with one, which calculates the TAX at 5% of the TOTAL, but only if sales are greater than or equal to 8,000. If the sales are less than 8,000 insert 0.

4. Replace the previous formula with the one that calculates the TAX for WHISKY at 8% of the TOTAL, and GIN at 6% of the TOTAL, but no other drinks require tax to be paid. For the drinks not requiring tax to be paid insert "No Tax".

5. Replace the previous formula with one, which calculates TAX using the Tax bands with the appropriate TAX RATES that you are provided.

6. DO NOT REMOVE THE PREVIOUS FORMULA.

7. Using the SUMIF function, calculate the TAX under Tax Band A in cell E14.

8. Calculate the TAX paid for the remainder of the Tax Bands.

Notes – Module 3

Notes – Module 3

Notes – Module 3

Notes – Module 3

MODULE 4
TEMPLATES

To create new workbooks that already have your customised formatting, you can base the new workbooks on a template. A template is a workbook that contains specific content and formatting so that you can use it as a basis for other similar workbooks. A template can include standardised text such as page headers and row and column labels, number formatting, Visual Basic macros, and custom toolbars.

You can save a workbook as a template that you can use to create other workbooks. In Microsoft Office Excel 2007, a template file (.xltx) can include data and formatting, and a macro-enabled template file (.xltm) can also include macros.

Below is a list of options that are stored in a template.

- The number and type of sheets in a workbook.
- Cell and sheet formats you set by using the commands on the Format menu.
- Cell styles.
- Page formats and print area settings for each sheet.
- Text you want to repeat in each new workbook or worksheet, such as page headers and row and column labels.
- Data, formulas, graphics, and other information you want each new workbook or worksheet to contain.
- Macros, hyperlinks, and ActiveX controls on forms. To make a custom toolbar available in new workbooks that you base on a template, attach the toolbar to the template.
- Protected and hidden areas of the workbook; for example, you can hide sheets, rows, and columns and prevent changes to worksheet cells.
- Workbook calculation options and window display options you set with Tools/Options.

CREATING

1. Open the workbook that you want to use as a template.

2. Click the Microsoft Office Button , and then click Save As.

3. In the File name box, type the name that you want to use for the template.

4. In the Save as type box, click Excel Template, or click Excel Macro-Enabled Template if the workbook contains macros that you want to make available in the template.

5. Click Save.

6. The template is automatically placed in the Templates folder to ensure that it will be available when you want to create a new workbook based on the template.

USING A TEMPLATE

1. Click the Microsoft Office Button , and then click New.

2. Under Templates, do one of the following:

3. To base the workbook on a recently used template, double-click the template under the Blank and Recently used Templates section.

4. To base the workbook on an installed template, click Installed Templates, and then under Installed Templates, double-click the template that you want to use.

5. To base the workbook on a template that you created, click My templates, and then on the My Templates tab, double-click the template that you want to use.

6. To base the workbook on another workbook, click New from Existing, and then double-click the workbook that you want to use.

EDITING A TEMPLATE

1. Open the template.

2. Make any amendments and click on Save.

NOTES – MODULE 4

Notes – Module 4

MODULE 5 - AUDITING A WORKBOOK

AUDITING AND WATCH WINDOW,

Auditing can help to trace errors, locate cells that refer to the current cell and attach comment notes to cells. Auditing can be used to troubleshoot errors as well as prevent errors by reviewing and commenting on your work.

Excels auditing tools allow you to do the following:

Trace Precedents	Cells that provide data for a formula
Trace Dependents	Formulas that refer to a cell
Remove Arrows	Remove Dependents, Precedents or all arrows

The auditing commands can be accessed from the Formula Auditing group in the Formulas Ribbon.
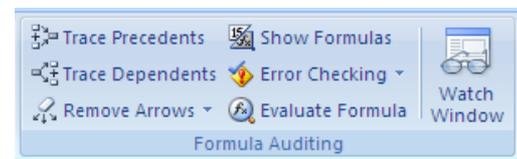
1. Click on the cell to audit.
2. Click onto Trace Precedents or Trace Dependents.

	Error Checking
	Trace Precedents
	Remove Precedent arrows
	Trace Dependents
	Remove Dependents Arrows
	Remove All Arrows

		Show Watch Window
		Evaluate Formula

CHECKING DATA FOR ERRORS

If a formula displays an error value such as #DIV/0!, you can use the Trace Error button to locate all cells referenced by the formula

Note: *The auditing arrows on the worksheet print out.*

FINDING DATA PRECEDENTS

1. Click on the cell that you want to audit.

2. Select Trace Precedents from the Formulas Ribbon.

3. Blue arrows point to the precedents. In the above example the precedents for cell D8 are shown.

FINDING FORMULA DEPENDANTS

1. Click on the cell that you want to audit.

2. Select Trace Dependents from the Formulas Ribbon.

3. Blue arrows point to the dependents. In the above example the dependents for cell D8 are shown.

WATCH WINDOW

The Watch Window holds important information, such as Book name, Sheet name, defined Name, Value, and Formula. Specific cells can be added so that their information can be viewed at any time.

Choose Watch window from the Formulas Ribbon.

Click Add Watch.

Click Add and choose the cells you want to watch.
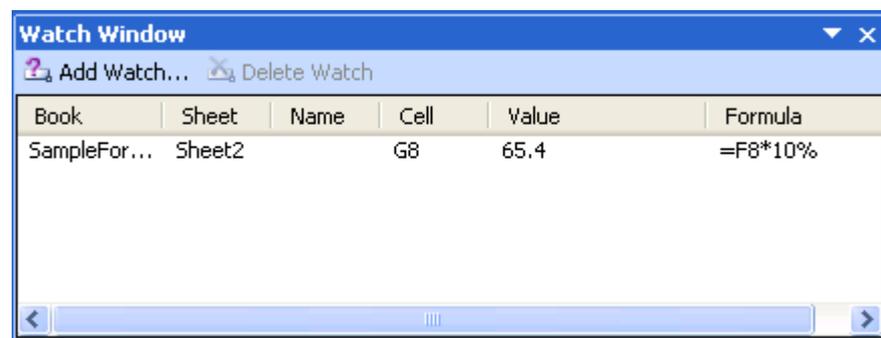
To display the cell that an entry in Watch Window toolbar refers to, double-click the entry.

Note: Cells that have links (external reference: A reference to a cell or range on a sheet in another Excel workbook, or a reference to a defined name in another workbook.) to other workbooks are displayed in the Watch Window toolbar only when the other workbook is open.

FORMULA AUDITING MODE

Excel uses certain rules to check for problems in formulas.

These can help you find common errors and correct them.

A triangle appears in the top-left corner of the cell when a problem is found. When you select the cell, a smart tag will appear.
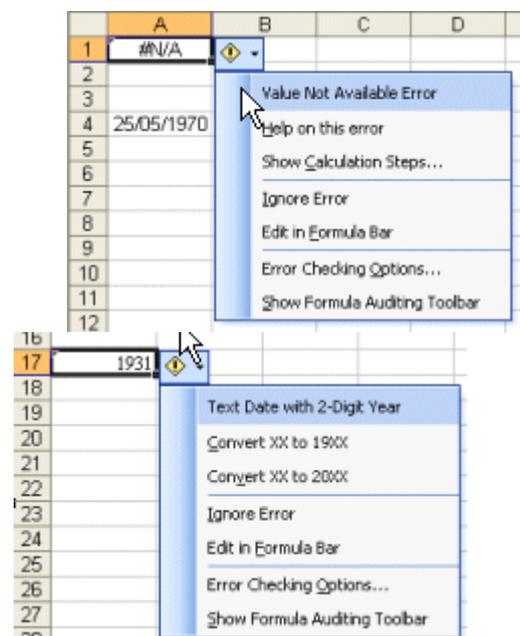
Click the down triangle to see the actions you can take.

There are several different rules that Excel checks for: •
Evaluates an error value –

The formula does not use the expected syntax, arguments, or data types.

Error values include #DIV/0!, #N/A, #NAME?, #NULL!, #NUM!, #REF!, and #VALUE!.

Each error value has different causes, and is resolved in different ways

Text date with 2 digit years

The cell contains a text date that can be misinterpreted as the wrong century when used in formulas. For example, the date in the formula =YEAR("1/1/31") could be 1931 or 2031. Use this rule to check for ambiguous text dates.

Number stored as text

The cell contains numbers stored as text.

These usually come from data imported from other sources. Numbers stored as text can cause unexpected sorting behaviours, and it is best to convert them to numbers.

Inconsistent formula in region.

The formula does not match the pattern of other formulas near it. In many cases formulas that are adjacent to other formulas only differ in the references used. If the references used in a formula are not consistent with those in the adjacent formulas, then the problem is noted.

Formula omits cells in region

The formula may not include a correct reference. If a formula refers to a range of cells, and you add cells to the bottom or right of that range, the references may no longer be correct. The formula does not always automatically update its reference to include the new cells. This rule compares the reference in a formula, against adjacent cells. If the adjacent cells contain more numbers (are not blank cells), then the problem is noted.

Unlocked cells contain formulas

The formula is not locked for protection. By default, all cells

are locked for protection, so the cell has been set to be unprotected. When a formula is protected it cannot be modified without being unprotected. Check to make sure you do not want the cell protected. Protecting cells that contain formulas prevents them from being changed, and can help avoid future errors.

Formulas refer to empty cells

The formula you are trying to use contains references to empty cells, which Excel will note, for your information.

NOTES – MODULE 5

Notes – Module 5

MODULE 6
DATA VALIDATION

Data Validation allows you to set limits to what data in a certain cell or range of cells. Criteria can be set to double check cell entries, display a message when a cell is selected or specify an error message when the entry does not meet the criteria.

	A	B	C	D
1	**Validation**			
2				
3	Company	No. of Shares	Price	Total
4	Premier	100	3.52	352
5	Books R Us	150	6.5	975
6	Lyons	70	2.3	161
7	Hamleys	180	8.9	1602
8				

SETTING DATA VALIDATION

The example above restricts the number of shares to a maximum value of 200. Data Validation has been set to cells B4:B7.

1. Select the cell(s) to validate.

2. Click on the Data ribbon and select Data Validation.
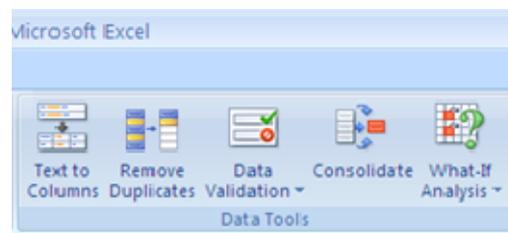
3. Click on the "Settings" tab.
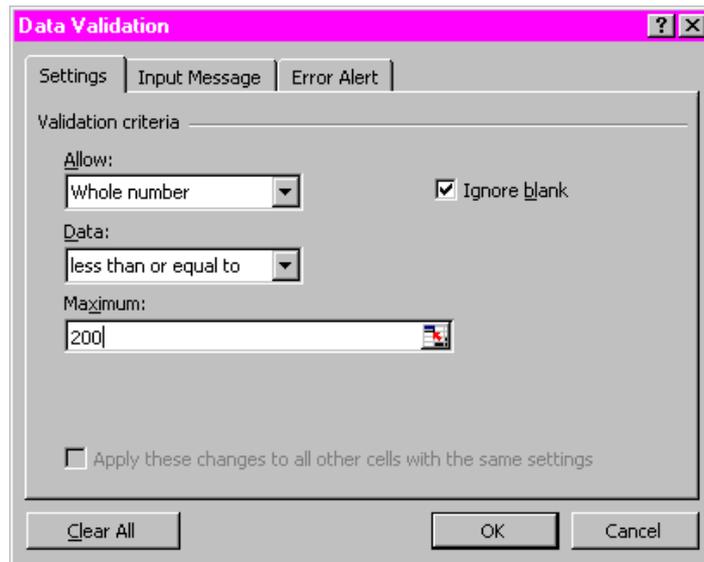
4. Change the "Allow" option to the constraints required. In this example they will be whole numbers and not decimals.

5. In the "Data" box select the comparative operator.

6. In the boxes below enter the maximum or minimum value.

7. Click on the "Input Message" tab. This options displays an input message reminding the user of the constraints that have been set to the cell(s).

8. Click on the "Error Alert" tab. An error message is displayed warning that invalid data has been inserted.
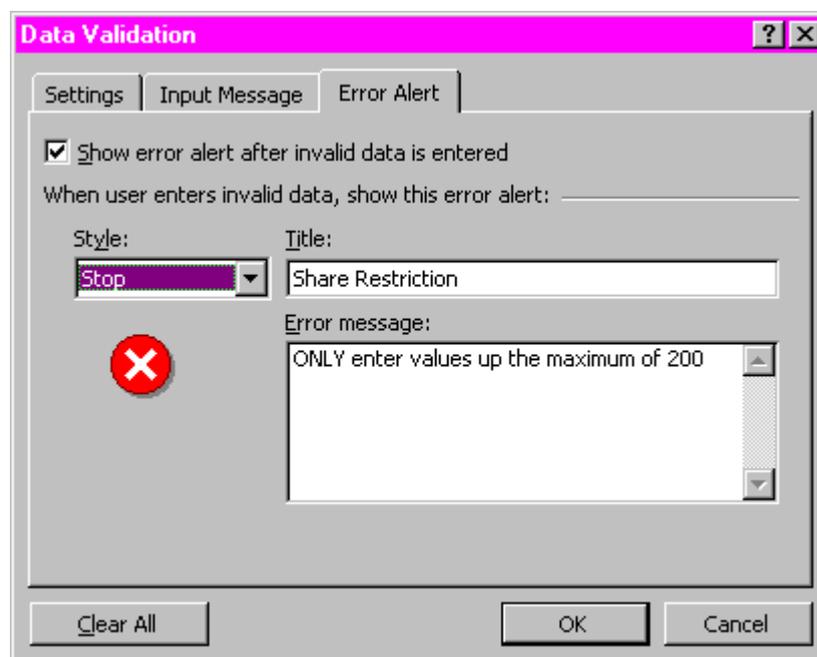
The "Style" of error message can be changed.

Stop   will prevent any invalid data being entered in the cell(s).

Warning   displays a warning message but invalid data can be entered.

Information   displays an information message allowing invalid data to be entered.

9. Type an error message and click on OK.

Share Restriction ☒

⚠ ONLY enter values up the maximum of 200

Continue?

[<u>Y</u>es] [<u>N</u>o] [Cancel]
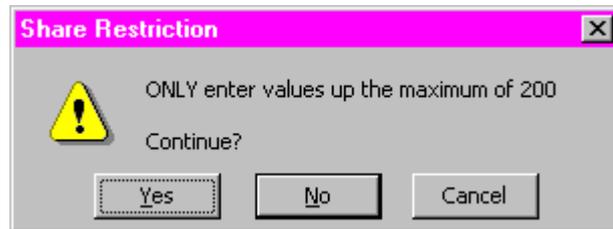
CHECKING FOR INVALID DATA

When you audit a worksheet for incorrect entries *Excel* can identify all cells that contain values that are outside the limits set by Data validation. These include values that are where typed in the cells, values that become incorrect because of calculations in formulas and values placed in cells by macros.

1. Click on the Data ribbon and select the Data Validation drop-down arrow.

2. Click on Circle Invalid Data. 🖩

	A	B	C	D
1	**Validation**			
2				
3	Company	No. of Shares	Price	Total
4	Premier	100	3.52	352
5	Books R Us	201	6.5	1306.5
6	Lyons	70	2.3	161
7	Hamleys	250	8.9	2225
8				
9				

3. Cells not matching certain validations are circled in red.

To Clear Validation Circles

1. Click on the Data ribbon and select the Data Validation drop-down arrow.

2. Click on Clear Validation Circles.

Notes – Module 6

Notes – Module 6

Notes – Module 6

Notes – Module 6

MODULE 7
MACROS

OVERVIEW OF MACROS/VBA

Macros are stored commands and functions that perform repeated tasks in Excel. A macro is stored in a Visual basic module that can be run whenever you need to perform a specific task.

Macros can be recorded or written using the programming language Visual Basic for applications. This manual covers recording macros and editing them using the Visual basic Editor. Premier Computer Solutions provide Visual Basic for Application courses.

Before recording or writing a macro, plan the steps and commands you want the macro to perform. If you make a mistake when recording the macro, any corrections you make will also be recorded. Each time you record a macro, the macro is stored in a new module attached to a workbook.

With the Visual Basic Editor, you can edit macros, copy macros from one module to another, copy macros between different workbooks, rename the modules that store the macros, or rename the macros.

To record a macro the Developer Ribbon needs to be added.

Adding the Developer Ribbon

1. Click the Microsoft Office Button , and then click Excel Options.
2. In the Popular category, under Top options for working with Excel, select the Show Developer tab in the Ribbon check box, and then click OK.

RECORDING A MACRO

1. Click on the Developer Ribbon and select Record Macro

2. In the "Macro name" box type a name for the macro.

The first character of a macro name must be a letter, other characters can be letters, numbers, or underscore characters. Spaces are not allowed.

3. To run the macro by using a keyboard shortcut, enter a letter in the "Shortcut key" box. You can use **CTRL+** letter (for lowercase) or **CTRL+SHIFT+** letter (for uppercase). The shortcut key will override any default *Excel* shortcut keys while the workbook that contains the macro is open.

4. In the "Store macro in" box, select the location where you want to store the macro.

5. For the macro to be available whenever you use *Excel*, store the macro in the Personal Macro Workbook in the XLStart folder, otherwise store in the active workbook or a different workbook.

6. Type a description for macro in the "Description" box. It is important that descriptions are entered as it makes it easier to refer back to macros at a later time.

7. Click on OK.

8. Carry out the actions you want to record.

9. Click onto Stop Recording.
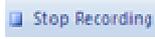
RUNNING MACROS

After you record a macro it can be run in Excel or from the Visual Basic Editor. You usually will run a macro in Excel; however, you can run the macro from the Visual Basic Editor while you are editing the macro. To interrupt the macro before it completes the actions you recorded, press ESC.

Running A Macro In Excel

Note: To interrupt a macro before it completes its actions, press ESC.

1. Open the workbook that contains the macro.

2. Click onto Macros from the Developer Ribbon.

3. Select the macro to run.

4. Click on the "Run" button.

Running A Macro From The Visual Basic Editor

Note: To interrupt a macro before it completes its actions, press ESC.

1. Open the workbook that contains the macro.

2. Click onto Macros from the Developer Ribbon.

3. Select the macro to run.

4. Click on the "Edit" button.

5. The macro code appears in a window.
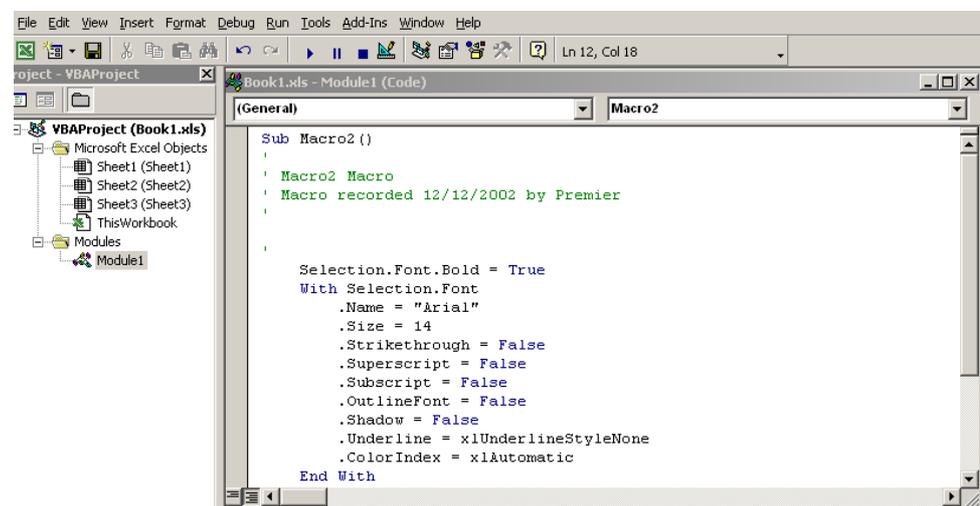
Note: If you want to run a different macro while you are in the Visual Basic Editor, click on the Tools menu and select Macros. Select the macro to run and then click on the "Run" button.

6. Click on the Run menu and select Run Sub/User Form.

7. Alternatively, click on the Run Sub/User Form button ▶ on the toolbar or press **F5**.

8. Closer the Visual Basic Editor once the macro has been run.

OVERVIEW OF THE VISUAL BASIC EDITOR

Visual Basic for Applications (VBA) is the standard macro language for Office 2007 products. The Visual Basic Editor is an application that you use to build VBA projects. Although the Visual Basic Editor appears to a completely separate application it uses the same memory as its host, in this case Excel. The Visual Basic Editor allows you to write code, build forms and run and test your application. Premier Computer Solutions provide dedicated VBA courses for all Office 2007 products.

Copying Macros

If you have an existing macro that contains commands you want to use in another macro, you can copy all or part of the macro to another module. You can also make a copy of a macro module to duplicate all macros stored in it.

Copy Part Of A Macro To Create Another Macro

1. Open the workbook that contains the macro you want to copy.

2. Click onto Macros from the Developer Ribbon.

3. Select the macro to run.

4. Click on the "Edit" button.

5. In the Visual Basic Editor, select the lines of the macro you want to copy.

6. To copy the entire macro, make sure to include the Sub and End Sub lines in the selection.

7. Click on Copy.

8. Switch to the module (window) where you want to place the code.

9. A module is a window that contains the VBA code for the macro.

10. Click on Paste.

ADD MACROS TO QUICK ACCESS TOOLBAR

1. Click the arrow to the right of the Quick Access Toolbar

2. Select More Commands.

3. Select Macros from the Choose command section.

4. Click onto the Macro name.

5. Click onto Add

6. Select OK

Assigning Macros To Graphical Objects

Macros can be assigned to buttons, drawing objects, or a graphic control on a worksheet. When you click on the button, drawing object or change the graphic control the macro will run automatically

Graphic controls are text box, list box, option button, or command button, that you place on a worksheet or form to display or enter data, perform an action, or make the worksheet or form easier to read.

1. Draw the object or graphical control.

2. Click with the right-mouse button on object or control.

3. From the sub menu choose "Assign Macro".

4. Select the macro to assign and click on OK.

SIMPLE EDITING OF MACROS

Editing Macros

Before you edit a macro, you should be familiar with the Visual Basic Editor. The Visual Basic Editor can be used to write and edit macros attached to Excel workbooks.

1. Open the workbook that contains the macro you want to edit.

2. Click onto Macros from the Developer Ribbon.

3. Select the macro to run.

4. Click on the "Edit" button.

5. Make the changes and then close the Visual Basic Editor or return back to the worksheet.

Note: All text proceeded by an apostrophe in the Visual Basic editor will appear "Green". These are Comments, which can be used to help explain the code.

EXERCISE – MODULE 7

1. Create a new macro called PRINT AREA, storing it in the current workbook.

2. Record a macro that is going to set a print area on your worksheet.

3. Add the macro to the Quick Access toolbar and also a macro button on the worksheet.

NOTES – MODULE 7

Notes – Module 7

Notes – Module 7

Notes – Module 7

MODULE 8
EXCEL'S ANALYTICAL TOOLS

GOAL SEEK

Goal Seeker allows you to perform simple forecasting in your worksheets. You can find a specific value for a defined result by adjusting the value of other cells in the worksheet. For example how many books you need to cell to generate an income of £200.

For the Goal Seeker to work the variable you want to adjust must be a formula and the formula must refer to other cells in the worksheet. The cell containing the formula is called the Set Cell.
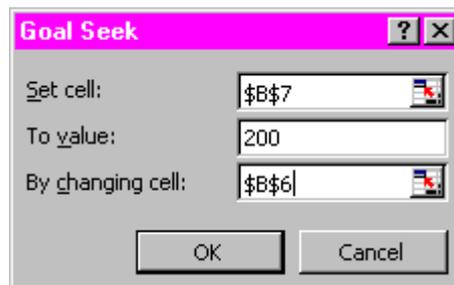
	A	B	C
1	Goal Seeker		
2			
3	Sales Forecast	for Books	
4			
5	Average Price	£6.99	
6	Sold	10	
7	Total Sales	£69.90	
8			

In the example above we need to adjust the number of books sold to reach a sales target of £200.

1. From the Data Tools group in the Data Ribbon click onto the drop down arrow next to What-if Analysis.

2. Select Goal Seek.
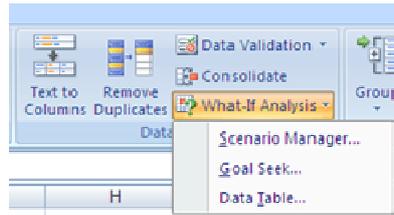
3. Click on the "Set Cell" containing the formula. In this example it is cell B7.

4. In the "To value" box type in the value you want to achieve.

5. In the "By changing cell" box select the cell containing the value that needs changing. In this example it is cell B6, the number of books sold.

6. Click on OK.

7. The results are displayed the Goal Seek Status dialog box.

8. Click on OK.

Note: To return back to the original data use Undo.
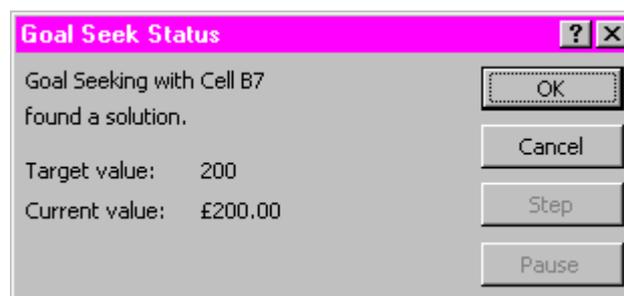
	A	B	C
1	**Goal Seeker**		
2			
3	**Sales Forecast for Books**		
4			
5	Average Price	£6.99	
6	Sold	28.6123	
7	Total Sales	£200.00	
8			

The result is that you will have to sell 29 books to achieve the target of £200.

SCENARIOS

A scenario is a set of values you use to forecast the outcome of a worksheet model. You can create and save different groups of values on a worksheet and then switch to any of these new scenarios to view different results.

Excel has a function called the Scenario Manager which enables you to create and save different scenarios and view them one at a time.

Creating A Scenario

	A	B	C	D
1	Scenario Manager			
2				
3	Sales and Commission on Training Material			
4				
5		Books	CBT	Videos
6	Average Price	£6.99	£60.00	£9.99
7	No.Sold	100	10	80
8	Commission Rate	2%	10%	7%
9	Total Sales	£699.00	£600.00	£799.20
10	Total Commission	£13.98	£60.00	£55.94
11				
12				

In the above example is the commission for the number of training material sold. The commission is based upon the number of books, CBT and videos sold. We are going to do a What-If analysis on the number of items sold.
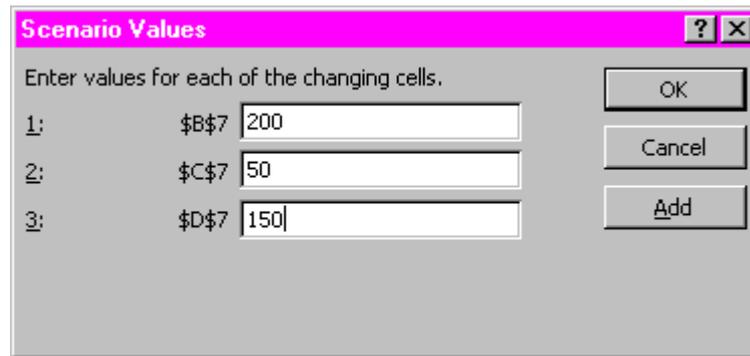
1. Select the cells that will change in the scenario. In this case cells B7:D7.

2. From the Data Tools group in the Data Ribbon click onto the drop down arrow next to What-if Analysis.
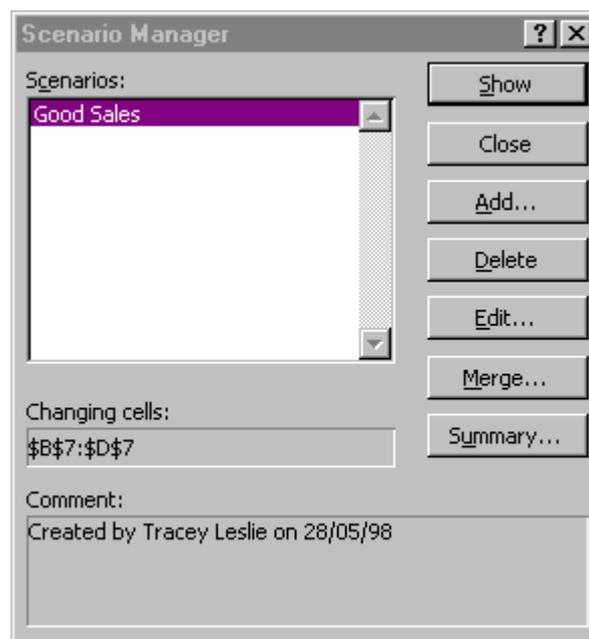
3. Select Scenario Manager.
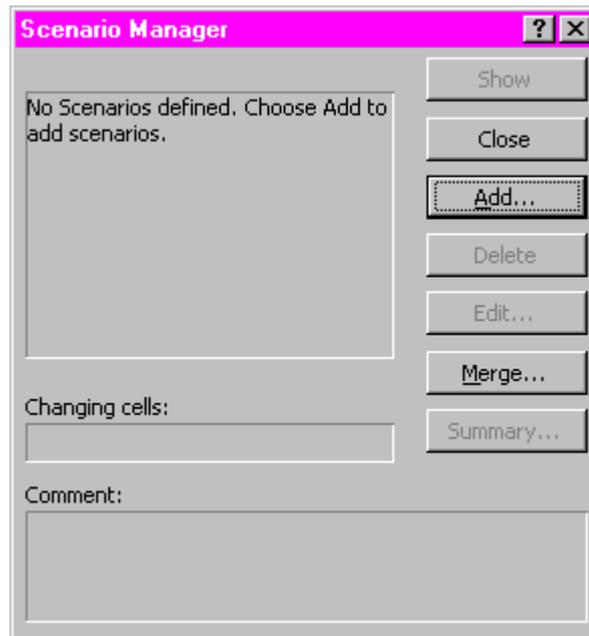
4. Click on the "Add" button.
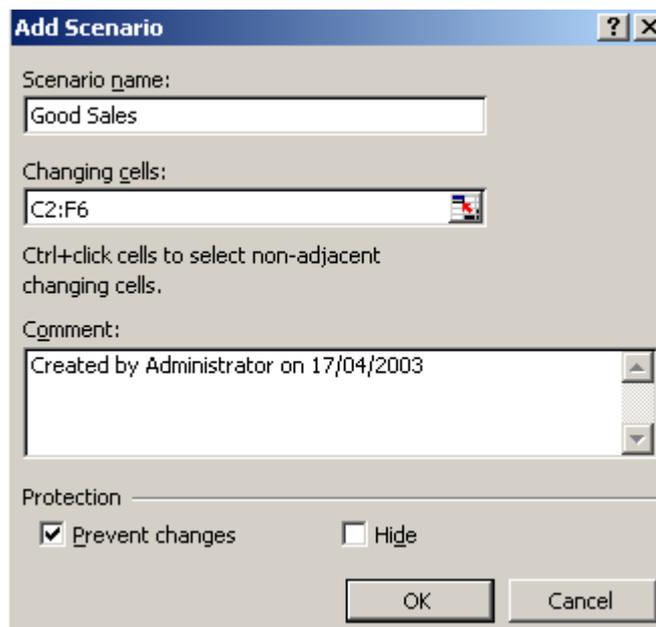
5. Type a name for the scenario.

6. In the "Changing cells" box, change the range of cells or cell that will change with the new values for the scenario, if need be.

7. Type any additional information in the "Comment" window.

8. Click on OK.

9. The "Scenarios Value" dialog box is displayed, this where you enter the data for each changing cell in the scenario.

10. Click on OK.

11. The "Scenario Manager" dialog box reappears. The new scenario appears in the Scenario window.

12. When you select a scenario the "Changing cells" box will display the cells to change.

13. To view the scenario, select the scenario name and click on the "Show" button.

14. To add a new scenario click on the "Add" button and repeat steps 2-8.

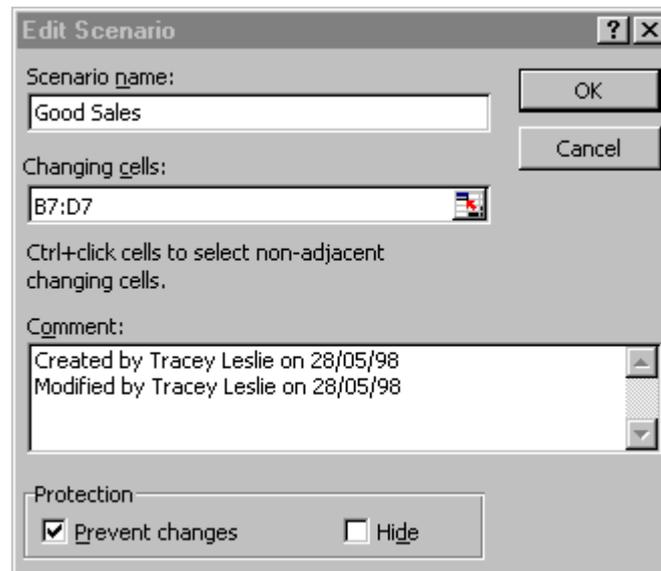15. Click on Close to return back to the worksheet.

Displaying A Scenario

1. From the Data Tools group in the Data Ribbon click onto the drop down arrow next to What-if Analysis.

2. Click onto Scenario Manager and select the scenario to view.

3. Click on the "Show" button.

4. The new values will be displayed in the worksheet.

5. Click on Close.

Editing Scenarios

1. From the Data Tools group in the Data Ribbon click onto the drop down arrow next to What-if Analysis.

2. In the Scenario Manager dialog box, select the scenario to change.

3. Click on the "Edit" button.

4. The "Edit scenario" dialog box opens.

5. Make any changes in the text boxes and click on OK.

6. Enter the new values in the "Scenario Values" dialog box.
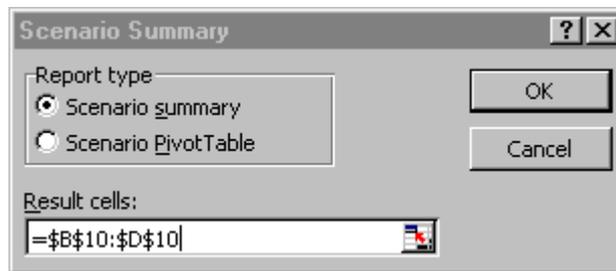
7. Click on OK.

Deleting A Scenario

1. From the Data Tools group in the Data Ribbon click onto the drop down arrow next to What-if Analysis.

2. Select the scenario to delete.

3. Click on the "Delete" button.

Summarising Scenarios

Excel provides two methods of displaying scenarios in a report. The Scenario Summary creates a simple report in a table form, showing the data for the changing cells and their effect on the results of formulas in a range. The second method is to generate a Pivot Table Summary from a multiple scenario set.

Creating A Summary Report

1. From the Data Tools group in the Data Ribbon click onto the drop down arrow next to What-if Analysis.

2. Select Scenario Manager.

3. Click on the "Summary" button.

4. Select "Summary Scenario" and enter the range of cells that contain formulas based on the input cells. In this example how the scenario affect the amount of commission.

5. Click on OK. *Excel* displays a new sheet with a summary table of the scenario inputs and results.

Scenario Summary		Current Values:	Good Sales	Current	Worst	Above Average
Changing Cells:						
	$B$7	100	200	100	20	120
	$C$7	10	50	10	3	15
	$D$7	80	150	80	30	90
Result Cells:						
	$B$10	£13.98	£27.96	£13.98	£2.80	£16.78
	$C$10	£60.00	£300.00	£60.00	£18.00	£90.00
	$D$10	£55.94	£104.90	£55.94	£20.98	£62.94

Notes: Current Values column represents values of changing cells at time Scenario Summary Report was created. Changing cells for each scenario are highlighted in gray.

Creating A Scenario Pivot Table Report.

1. From the Data Tools group in the Data Ribbon click onto the drop down arrow next to What-if Analysis.
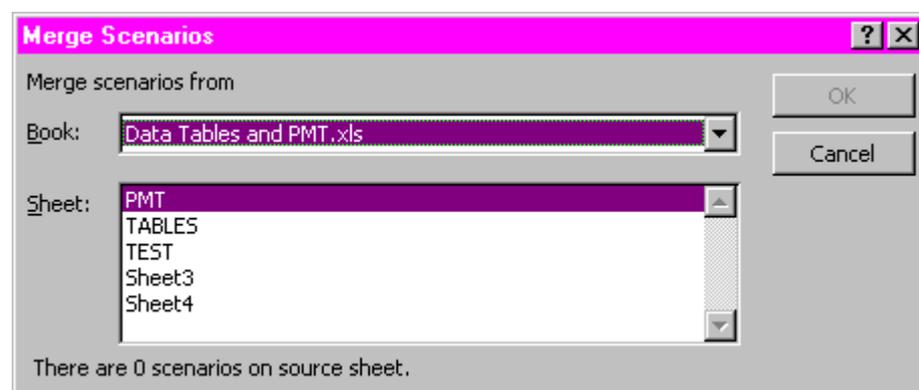
2. Select Scenario Manager

3. Click on the Summary button.

4. Select "Scenario Pivot Table" and enter the range of cells that contain formulas based on the input cells. In this example how the scenario affect the amount of commission.

5. Click on OK. *Excel* displays a new sheet with a pivot table of the scenario inputs and results.

	A	B	C	D
1	$B$7:$D$7 by	(All)		
2				
3		Result Cells		
4	$B$7:$D$7	$B$10	$C$10	$D$10
5	Above Average	16.776	90	62.937
6	Current	13.98	60	55.944
7	Good Sales	27.96	300	104.895
8	Worst	2.796	18	20.979
9				

Merging Scenarios

It is easier to merge scenarios when all what-if worksheets in workbooks are the same.

Note: You can add scenarios when a worksheet is protected, but you cannot edit or delete existing scenarios while a worksheet is protected unless you clear the Prevent changes check box in the Edit Scenario dialog box. You can still edit the values in changing cells on the worksheet unless the cells are locked.

To remove a scenario from the list in the Scenario Manager dialog box, select the Hide check box.

1. From the Data Tools group in the Data Ribbon click onto the drop down arrow next to What-if Analysis.

2. Select Scenario Manager.

3. Click on the "Merge" button.

4. In the "Book" box select the workbook that you want to merge the scenarios from.

Merge Scenarios

Merge scenarios from

Book: Data Tables and PMT.xls

Sheet: PMT
TABLES
TEST
Sheet3
Sheet4

OK Cancel

There are 0 scenarios on source sheet.

5. In the "Sheet" box select the names of the worksheets that

contain the scenarios you want to merge. All changing cells on the source worksheets must refer to the corresponding changing cells on the active worksheet. *Excel* copies all scenarios on the source sheets to the active worksheet.

6. Click on OK.

SOLVER

You can also determine resulting values when you need to change more than one cell used in a formula and have multiple constraints for those values. Solver adjusts the values in the cells you specify to produce the result you want from the formula.

Solver determines the optimal advertising budget for each department, cells B7:E7 without exceeding the £40,000 total of those cells

	A	B	C	D	E	F
1						
2		Q1	Q2	Q3	Q4	Totals
3	Units Sold	3,592	4,390	3,192	4,789	15,963
4						
5	Sales Revenue	£143,662	£175,587	£127,700	£191,549	£638,498
6	Cost of Sales	£89,789	£109,742	£79,812	£119,718	£399,061
7	Advertising	£7,273	£12,346	£5,118	£15,263	£40,000
8						
9	Overheads	£21,549	£26,338	£19,155	£28,732	£95,775
10	Profit	£22,324	£26,338	£19,155	£33,099	£100,916
11						
12	Product Price	£40.00				
13	Product Cost	£25.00				
14						

The final result is the largest profit given the constraint on the advertising budget

For example, use Solver to maximize the profit shown in cell F10 by changing the quarterly advertising budget (cells B7:E7) while limiting the total advertising budget (cell F7) to £40,000.

Loading the Solver

Note: Any Add-ins you select in the Add-Ins dialog box remain active until you remove them..

The Solver Add-in is a Microsoft Office Excel add-in program that is available when you install Microsoft Office or Excel. To use it in Excel, however, you need to load it first.

1. Click the Microsoft Office Button , and then click Excel

Options.

2. Click Add-Ins, and then in the Manage box, select Excel Add-ins.

3. Click Go.

4. In the Add-Ins available box, select the Solver Add-in check box, and then click OK.

5. Tip If Solver Add-in is not listed in the Add-Ins available box, click Browse to locate the add-in.

6. If you get prompted that the Solver Add-in is not currently installed on your computer, click Yes to install it.

7. After you load the Solver Add-in, the Solver command is available in the Analysis group on the Data tab.

Define and solve a problem by using Solver

1. Click onto Solver from the Data Ribbon.

2. In the "Set Target Cell" box, enter a cell reference or name for the target cell. The target cell must contain a formula.

   To have the value of the target cell be as large as possible, click on Max.

   To have the value of the target cell be as small as possible, click **Min**.

   To have the target cell be a certain value, click **Value of**, and then type the value in the box.

3. In the "By Changing Cells" box, enter a name or reference for each adjustable cell, separating nonadjacent references with commas. The adjustable cells must be related directly or indirectly to the target cell. You can specify up to 200 adjustable cells.

   To have Solver automatically propose the adjustable cells based on the target cell, click **Guess**.

4. In the "Subject to the Constraints" box, enter any constraints you want to apply.

5. Click on Solve.

6. To keep the solution values on the worksheet, click Keep Solver Solution in the Solver Results dialog box.

   To restore the original data, click on Restore Original Values.
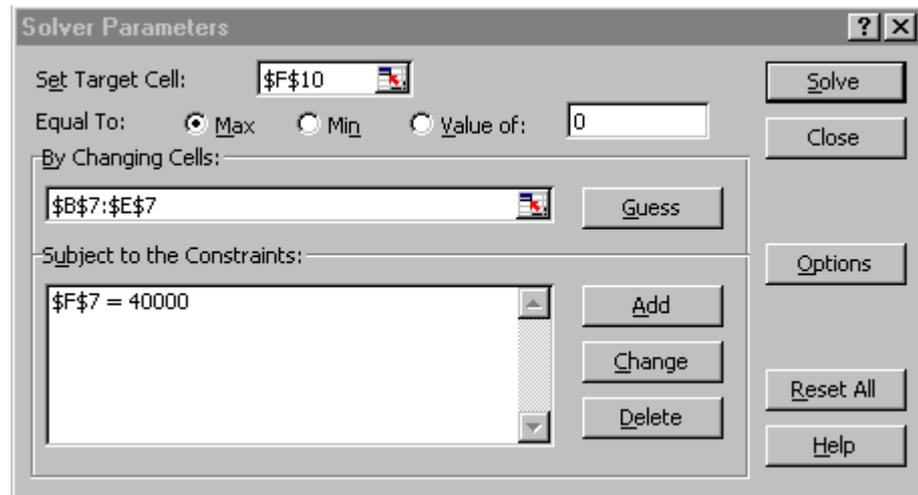
2 & 1 WAY INPUT TABLE

A data table is a range of cells that display how changing certain values in your formulas affect the results of the formulas. Data tables provide a shortcut for calculating multiple versions in one operation and a way to view and compare the results of all of the different variations together on your worksheet.

There are two types of data tables, a one-variable data table or two-variable data table.

One-Variable Data Table

A one-variable data table allows you to determine a result to a different variable based on the same calculation

For example, how different interest rates affect a monthly mortgage payment.

	A	B	C	D	E	F
1	One Input Data Table					
2				Different Interest Rates	Monthly Payment	
3					-£981.39	
4	Loan Amount	125,000		8.75%		
5	Interest	8.20%		9.00%		
6	Term	25		9.10%		
7	Loan Repayment	-£981.39		9.20%		
8				9.50%		
9				10.00%		
10						
11						

In the example above in cell B7 the function

=PMT(B5/12,B6*12,B4) has been used to calculate the monthly repayment at 8.2%.

In column D are the different interest rates that we want to use in the =PMT() function. Column E is where the results will be displayed.

1. Reference the cell containing the formula to a cell at the top of the column where the results will be displayed.

2. To do this in cell E3 press equal (=), click on cell B7 and then press ENTER.

3. Select the cells to be used in the data table. In this case cells D3:E9.

4. From the Data Tools group in the Data Ribbon click onto the drop down arrow next to What-if Analysis.

5. Select DataTables.

6. In the "Column input cell" box select the cell used in the original calculation (in cell B7) which will be the variable factor. In this case it is the interest rate in cell B5.
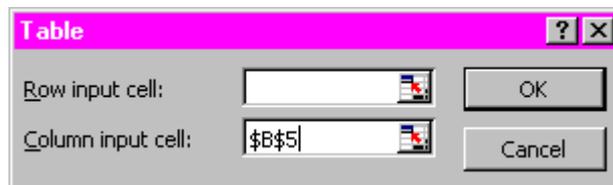
7. Click on OK.

Note: You must design one-variable data tables so that input values are listed either down a column (column-oriented) as above or across a row (row-oriented). Formulas used in a one-variable data table must refer to an input cell.

	A	B	C	D	E	F
1	**One Input Data Table**					
2				**Different Interest Rates**	**Monthly Payment**	
3					-£981.39	
4	Loan Amount	125,000		8.75%	-£1,027.68	
5	Interest	8.20%		9.00%	-£1,049.00	
6	Term	25		9.10%	-£1,057.57	
7	Loan Repayment	-£981.39		9.20%	-£1,066.17	
8				9.50%	-£1,092.12	
9				10.00%	-£1,135.88	
10						

Two-Variable Data Tables

Two-variable input tables calculate two variable factors, for example a different interest rate and a different loan amount.

	A	B	C	D	E	
1	Two-Variable Data Table					
2						
3						
4	Loan Amount	125,000				
5	Interest	8.20%				
6	Term	25				
7	Loan Repaymen	-£981.39				
8						
9						
10	Loan Amounts	Interest Rates				
11	-£981.39	8.75%	9%	9.25%	9.50%	
12	175,000					
13	200,000					
14	225,000					
15	250,000					
16						

In the example above in cell B7 the function

> =PMT(B5/12,B6*12,B4) has been used to calculate the monthly repayment at 8.2%.

In row 11 are the different interest rates and in cells A12:A15 are the different loan amounts.

8. Reference the cell containing the formula to a cell at the top of the column where the table starts and the results will be displayed.

9. To do this in cell A11 press equal (=), click on cell B7 and then press ENTER.

10. Select the cells to be used in the data table. In this case cells A11:E15.

11. From the Data Tools group in the Data Ribbon click onto the drop down arrow next to What-if Analysis.

12. Select Data Table.

13. In the "Row input cell" box select the cell used in the original calculation for the interest rate (cell B5) which will be one of the variable factors. The interest rates are displayed across a row, if they were going down a column then use "Column input cell".

> In the "Column input cell" box select the cell used in the original calculation for the loan (cell B4). The loan amounts are displayed down a column, if they were going across a row then use "Row input cell".

14. Click on OK.

	A	B	C	D	E
1	Two-Variable Data Table				
2					
3					
4	Loan Amount	125,000			
5	Interest	8.20%			
6	Term	25			
7	Loan Repaymen	-£981.39			
8					
9					
10	Loan Amounts	Interest Rates			
11	-£981.39	8.75%	9%	9.25%	9.50%
12	175,000	-£1,438.75	-£1,468.59	-£1,498.67	-£1,528.97
13	200,000	-£1,644.29	-£1,678.39	-£1,712.76	-£1,747.39
14	225,000	-£1,849.82	-£1,888.19	-£1,926.86	-£1,965.82
15	250,000	-£2,055.36	-£2,097.99	-£2,140.95	-£2,184.24
16					

NOTES – MODULE 8

Notes – Module 8

Notes – Module 8

Notes – Module 8

INDEX

P R E M I E R

Consultancy and Training Services

**Management ♦ Personal Development ♦ Software ♦ Technical
♦ e-Learning ♦ e-Business ♦ Business Integration ♦**

Premier offers a comprehensive range of services including

Full training needs analysis	Consultancy
Tailored courses	Seminars
Scheduled course programme	Workshops
On-site training	'One to one' coaching
Training audits	e-Learning

How to find Premier's City training centre
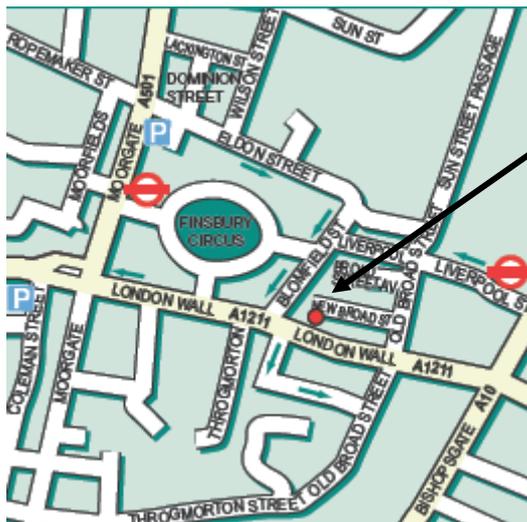
We are here:
Maven Centre
3rd Floor, 42 New Broad Street
London
EC2M 1SB

Bookings and enquiries:

Telephone +44 (0) 20 7729 1811
Fax +44 (0) 20 7729 9412
Email enquiries@premiercs.co.uk
Website www.premcs.com

Microsoft®
C E R T I F I E D
Partner