

Using Microsoft Excel® Functionality to Promote Academic Integrity in the Business Classroom

Mary Funck and Ashly Smith

The authors of this paper have developed a technique in Microsoft Excel® that easily allows instructors to create assignments customized for each student in a class. Using assignments that vary input values and/or question prompts serves as a deterrent to assignment ‘over-sharing’ behaviors on the part of students.

INTRODUCTION

Assignments and exams in academic courses are generally designed to assess students’ mastery of course content. To create an assessment that will accurately measure a student’s knowledge and ability, it is often necessary for professors to incorporate in the assessment activity, safeguards against academic dishonesty. Preventing and detecting academic dishonesty may be particularly relevant in business courses given that, despite a common emphasis on ethical behavior in business curricula, some survey research shows that 75% of surveyed Master of Business Administration (MBA) students admit to cheating in order to gain admission to their graduate programs (Cardon, 2018, p. 5).

Computer software programs provide several options for detecting various forms of academic dishonesty. These programs, which are often available within learning management systems (LMS), identify inappropriate behavior by reviewing assignment submissions for plagiarism and video recording students while they complete tests.

An alternate approach of promoting academic integrity focuses on the *prevention* of unethical behavior rather than on its detection (Michael & Williams, 2013). For example, LMS, like Blackboard, can deter academic dishonesty by blocking students from viewing prohibited resources during a test period. “Locking down” the testing environment, however, often disallows students’ access to resources that are essential to the completion of an assessment activity

LMS also discourage academic dishonesty by offering the capability to randomize the question text or numeric inputs used in assignment/test problems. Some instructors, however, do not feel competent to or face time constraints in setting up the sometimes-complicated rules and systems needed to create these randomized questions in an LMS. In addition, some instructors would rather assign a test or project outside of the LMS in order to help students more effectively learn a necessary software program—like Microsoft Excel—or because another program offers better opportunities to practice the course material.

Unfortunately, students’ creativity and ingenuity allow them to develop techniques to circumvent nearly every

strategy implemented by professors for the prevention and detection of behaviors associated with academic dishonesty. One such strategy that students use to “help” each other earn high scores on assessment activities is to share an activity’s questions and solutions. Within this strategy, one student completes the assessment, records the questions, and then shares the questions and solutions with other students—sometimes even saving the question/answer information in a database or posting it to online sites for students who enroll in later sections of the course. For assessments that involve the submission of an electronic file, such as a Microsoft Word or Excel file, this “sharing” of an assessment solution can be accomplished easily by copying a file and modifying a few of its properties.

The technique introduced in this article attempts to discourage academic dishonesty by using Excel functionality to present each student completing an assessment (assignment or test) with a different set of numerical inputs to common questions and/or by modifying the assignment tasks presented to each student (thus producing unique resulting solutions). Although this technique can aid in the detection of academic dishonesty, it was developed with the understanding that providing unique assignments to each student increases the effort required for students to “share” solutions and thus encourages students to complete independently their own assessment activities.

We use a simplified example to demonstrate this technique. The sample assessment involves an assignment that

- (1) Presents unique inputs to a calculated problem for each student (on worksheet #1 of our example file), and
- (2) Asks students to solve a different tasks - one version of the assignment requests the calculation of the future value of a financial instrument while another version instructs a student to compute the present value (on worksheet #2 of our example file).

The technique used in this simple example can easily be expanded to create assessments that are more complex and substantially different for each student. In addition, this demonstration utilizes an assessment completed by students solely within Excel; it could also be used to generate unique

instructions or assessment instruments that are saved to PDF files or are printed for distribution.

METHODOLOGY

The following six steps can be followed to implement this technique.

Within the assignment (Excel) workbook:

1. Create a separate worksheet that lists in consecutive worksheet rows, the name of each student and the corresponding unique values (i.e. numerical inputs or question text). I refer to this worksheet as the ‘Information’ worksheet. In the image below, note that student names are listed in column A, and unique values for each student are listed (by row) in columns B through H.

The unique values in columns B through D will be used on worksheet #1 to provide different inputs to common tasks for each student. In Figure 1, note that student *Linsay* (worksheet row 2) has a ‘Widget Increment’ of 25 (column B) where student *Karter’s* (worksheet row 3) widget increment is 30.

The unique values in columns E through H will be used on worksheet #2. Values in column E will instruct students to complete the present or the future value of a financial instrument, whose characteristics are determined by the values in columns F through H.

Figure 1: ‘Information’ Worksheet Values

Student	Widget Increment	Price Increment	Tax Rate	PV/FV	PV/FV Value	Investment Horizon	APR
Linsay	25	0.50	0.32	Present	3600	15	5.25
Karter	30	0.75	0.34	Future	3800	25	5.35
Parker	35	0.35	0.36	Present	3950	17	5.45
Ryan	40	0.50	0.38	Present	2950	20	5.65
Travis	45	0.65	0.42	Future	2975	22	5.75
Danielle	50	0.25	0.40	Future	4560	14	5.25

2. Modify the assignment workbook by adding worksheet(s) that contain the generic (non-dynamic) portion of the assignment. The content can encompass one or many worksheets - it need only contain opportunities for substituting unique values or question statements for each student. Note that in the figures 2A and 2B, only the text that is consistent for all students is entered into the worksheet at this point of the assignment completion process. The characters ‘***’ have been included in the image to indicate the position at which the dynamic values/text will later be incorporated.

Figure 2: Generic Worksheet #1 & #2 Values

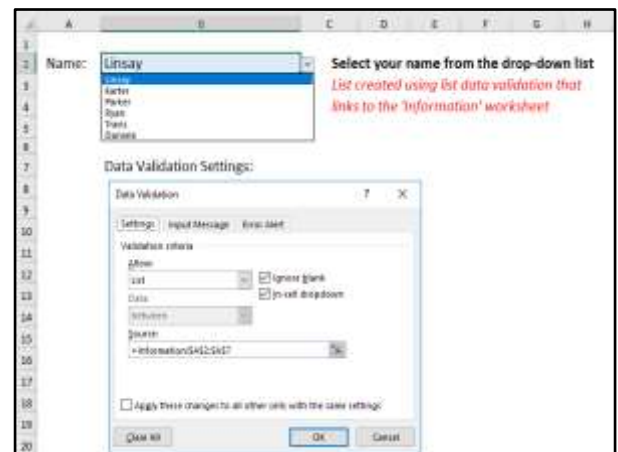
Figure 2A: Worksheet #1 generic content:

Figure 2B: Worksheet #2 generic content:

Frequency	Periods/Year	***	***
Annual			
Semiannual			
Quarterly			
Monthly			
Weekly			
Daily			

3. Modify the assignment workbook by adding an ‘Instructions’ worksheet. On this worksheet, use Excel’s list data validation to allow each student to select his or her name as the assignment’s author (cell B2). The student’s name becomes the ‘lookup’ value used to identify the correct row of unique values on the ‘Information’ worksheet. Requiring students to select their name from a list created using data validation is essential, as the name selected on the ‘Instructions’ worksheet must exactly match the list of student names in the first column of data on the ‘Information’ worksheet.

Figure 3: ‘Instructions’ Worksheet Values



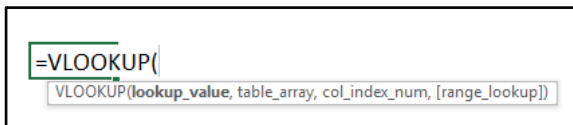
Note: Data validation is accessed through ‘Data Tools’ group on the ‘Data’ ribbon. The data validation setting for cell B2 on the ‘Instruction’ worksheet are shown in Figure 3.

4. Modify the assignment workbook by incorporating Excel’s CONCATENATE, VLOOKUP and TEXT functions – these functions work in tandem to insert the

unique, formatted values/question text into the assignment. The VLOOKUP function ‘looks up’ the values in the data region unique to the student completing the assignment (finds the correct row on the ‘Information’ worksheet). The TEXT function converts the unique numeric inputs into formatted text strings. Then, the CONCATENATE function joins the static (generic) portion of the question prompt with the formatted, dynamic portion to create a set of inputs or question prompt that is unique to each student.

For each of these functions, I include the function signatures and an explanation of each of the function arguments.

The VLOOKUP function matches a value in the first row of a vertical table structure and returns data contained in the matching row.

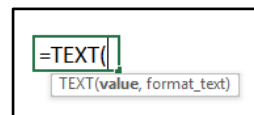


The arguments in the VLOOKUP function refer to the following:

- The *lookup_value* argument provides the value the VLOOKUP function will ‘match’ in the data region on the ‘Information’ worksheet (the area that contains the unique data values for each student). In our example, it is a reference to the student name selected in cell B2 of the ‘Instructions’ worksheet. The VLOOKUP function uses the name in cell B2 to determine the matching row of values on the ‘Information’ worksheet. For example, if *Linsay* is selected as the student name on the ‘Instructions’ worksheet, then the VLOOKUP function will match with the second row of the data region (worksheet row 3) on the ‘Information’ worksheet.
- The *table_array* refers to the location of the data region containing the student names and unique associated information. In our example, this data region is located on the ‘Information’ worksheet in range A2:H7.
- The *column_index_num* designates which column in the data region (*table_array*) should be returned by the function. Column numbers start with one for the leftmost column (of the data region) and increment by one for each consecutive column as you move to the right. For the data region in our example, a column index number of 3 would return the ‘Price Increment’ for the row matching the student’s name selected in cell B2 on the ‘Instructions’ worksheet.

- The optional *range_lookup* argument determines whether the VLOOKUP function identifies an exact match for the *lookup_value* (student name) or if matching a range of values is allowed. Our technique relies on an exact match for each student name; thus, the *range_lookup* value is always set to false.

The TEXT function is used to convert a numeric value to a formatted string. Note: with our method, the TEXT function is used only when incorporating dynamic *numeric* values into a question prompt; it is not used when incorporating text (such as ‘Present’ or ‘Future’ in column E of the ‘Information’ worksheet).



The arguments in the TEXT function refer to the following:

- The *value* argument is a number, cell reference, formula, or function that resolves to a numeric value. Our method nests the VLOOKUP function within the TEXT function. The *value* argument is then a numeric value returned by the VLOOKUP function after the function ‘finds’ the correct student (row) and variable (column). For example, if *Linsay* is selected as the student name (cell B2 of the ‘Instructions’ worksheet), and the VLOOKUP function has a column index number of 4, then the *value* argument of the TEXT function will be the tax rate of 0.32 located in cell D2 of the ‘Information’ worksheet that is returned by the VLOOKUP function.
- The *format_text* argument provides Excel a pattern for formatting the numeric *value* argument. Format patterns are enclosed in double quotes, and they use custom format characters to build a format pattern. Common custom format characters include the currency sign (\$), percentage symbol (%), commas, decimal points, the digit 0 (to display significant and non-significant numbers), and the character # to display only significant numeric values. For example, the format pattern “\$#,##0.00” display a numeric value as currency with two decimal places of precision, and a thousand separator for number >= 1,000.

The CONCATENATE function is used to join (merge) one or more text strings into a single string. This function has one required text string argument *text1* that can be combined with up to 254 additional optional strings [*text2*], The string arguments are separated within the function call by commas. The function’s

arguments can be hard-coded strings (i.e. “dog”), cell references, or function calls that return strings (i.e. VLOOKUP).



To incorporate the dynamic (unique) portion of the assignment, add the formulas shown in Exhibit 1 Panel A to worksheet #1. Notice how the static (generic) text added earlier is now incorporated within the CONCATENATE function and the input values (formerly represented by ‘***’) are now replaced with calls to the TEXT and/or VLOOKUP functions.

Figure 4: ‘Worksheet #1’ Dynamic Values

When the student *Linsay* is selected from the list in cell B2 of the ‘Instructions’ worksheet, worksheet #1 displays the following question prompt:

The number of units per order start at 100 and increase in increments of 25
 The price per unit begins at \$14.75 and decreases in increments of \$0.50
 The tax rate is 32%

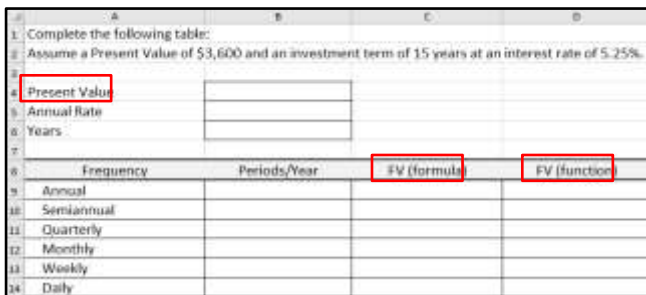
Alternatively, when the student *Karter* is selected from the list in cell B2 of the ‘Instructions’ worksheet, worksheet #1 displays the following question prompt:

The number of units per order start at 100 and increase in increments of 30
 The price per unit begins at \$14.75 and decreases in increments of \$0.75
 The tax rate is 34%

Worksheet #1 now provides each student with different inputs for common tasks. Worksheet #2 not only provides different inputs to question prompts, it also requires students to perform different calculations. Worksheet #2 can be modified by incorporating the Excel formulas shown in Exhibit 1 Panel B.

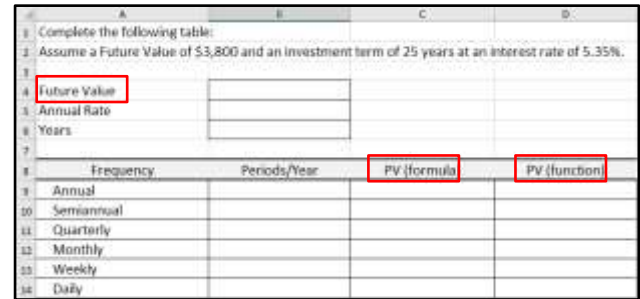
When the student *Linsay* is selected from the list in cell B2 of the ‘Instructions’ worksheet, worksheet #2 displays the following question prompt (Figure 5):

Figure 5: ‘Worksheet #2’ Values with Student Linsay



Compare the image above to the worksheet created when student *Karter* is selected from the list in cell B2 of the ‘Instructions’ worksheet (Figure 6). Not only are the numeric inputs listed in the instructions unique (cell A2), the first student is asked to calculate the future value (Figure 5A) and the second student the present value of the financial instrument.

Figure 6: ‘Worksheet #2 Values with Student Karter



Note: the labels in cell A4 is determined by the VLOOKUP function - the label matches the value in column E of the ‘Information’ worksheet.

The labels in cell C8 and D8 are determined by an IF statement with the following logic:

- If the student is provided with the present value (as with *Linsay’s* example), then they are asked to compute the future value in columns C and D.
- Else, if the student is provided with the future value (as with *Karter’s* example), they they are asked to compute the present value in columns C and D.

5. Now that the formulas have been added to the Excel file to create a dynamic assignment, worksheet level protection should be enforced to safeguard the code. Worksheet protection relies on both the lock/unlock status of the cell and the protected/unprotected status of the worksheet.

For each worksheet containing the dynamic statements:

- Select the ‘Select All’ button in the upper left corner of the works sheet.
- In the ‘Format Cells’ dialog box, select the ‘Protection’ tab, and ensure that both the ‘Locked’ and ‘Hidden’ check boxes are deselected. This action unlocks all cells for editing (even if worksheet protection is applied).

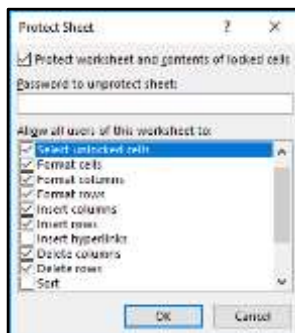
Figure 7: Cell Protection Settings



- Select all of the cells containing dynamic code (use the Control key to select non-continuous cells)
- In the 'Format Cells' dialog box, select the 'Protection' tab and check the 'Locked' check box. Selecting this option, when combined with worksheet protection, will prohibit students from modifying the content of locked cells.
- Check the 'Hidden' option if you would like the content of these cells to be hidden from the student. With this option selected, the contents of the cell, when selected, will not display in the formula bar.
- Apply worksheet protection through 'Changes' group on the 'Review' ribbon. The "Protect Sheet" dialog box allows the instructor to set which actions are allowed on locked cells. It is important to note that the instructor's selection in this area should reflect the capabilities necessary for student to complete the assignment tasks. Applying protection will make some Excel tasks unavailable to students; these tasks will be 'grayed-out' on the Excel ribbons when not available.

I recommend using a password so that students are not able to 'unprotect' the worksheet and modify the contents of the cells containing dynamic text.

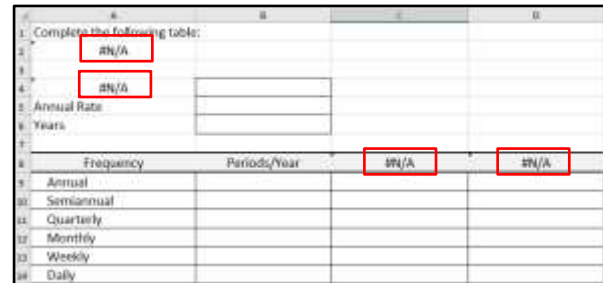
Figure 8: Worksheet Protection Settings



The assignment should be tested prior to its distribution to students to ensure that selecting each student names produce valid assignment text.

If the student name selected in cell B2 of the 'Instruction' worksheet does not match any of the student names in column A of the 'Information' worksheet, the "#N/A" symbol will display in cells containing the dynamic text (as shown in Figure 9).

Figure 9: Invalid Worksheet Output Values



	Frequency	Periods/Year		
Annual				
Semiannual				
Quarterly				
Monthly				
Weekly				
Daily				

6. The final step in this assignment creation process is to disguise the data on the 'Information' worksheet and then hide the 'Information' worksheet from student view. This step helps to maintain the integrity of the dynamic assignment by removing student access to the 'working parts' of this technique.

The disguise the data entered on the 'Information' worksheet, make the following changes to the cell range containing the student name and unique data values (Cells A1:H7 in our example):

- Change the fill color to 'No Fill'
- Change the font color to 'white'
- Hide the columns containing the data

To hide the worksheet from student view, right-click on the worksheet tab and select the 'Hide' option. Instructors also have the option of protecting the structure of a workbook to disallow student from 'un-hiding' hidden worksheets.

CONCLUSION

Enforcing academic integrity is vital in providing accurate and meaningful assessments in the college classroom. Through this article, the authors have presented one method that to create unique assessment instruments, which might potentially reduce dishonest 'over sharing' behavior on the part of students.

REFERENCES

- Cardon, P. W. (2018). Business communication: Developing leaders for a networked world. 3rd ed. New York, NY: McGraw Hill Education.
- Michael, T. B., & Williams, M. A. (2013). Student equity: Discouraging cheating in online courses. *Administrative Issues Journal: Education, Practice & Research*, 3(2), 30-41. DOI: 10.5929/2013.3.2.8

Mary Funck is an Associate Professor of Finance at Sam Houston State University.

Ashly Smith is an Assistant Professor of Business Communication at Sam Houston State University

EXHIBIT 1: Formulas for Inserting Dynamic Text

Panel A: Worksheet #1 Code

Cell B1: =CONCATENATE("The number of units per order start at 100 and increase in increments of ",VLOOKUP(Instructions!B2,Information!A2:H7,2,FALSE))
Cell B2: =CONCATENATE("The price per unit begins at \$14.75 and decreases in increments of ",TEXT(VLOOKUP(Instructions!B2,Information!A2:D7,3,FALSE),"\$0.00"))
Cell B3: =CONCATENATE("The tax rate is ",TEXT(VLOOKUP(Instructions!B2,Information!A2:H7,4,FALSE),"0%"))

Panel B: Worksheet #2 Code

Cell A2: ="Assume a "&VLOOKUP(Instructions!B2,Information!A2:H7,5,FALSE)&" Value of "&TEXT(VLOOKUP(Instructions!B2,Information!A2:H7,6,FALSE),"\$#,##0")&" and an investment term of "&VLOOKUP(Instructions!B2,Information!A2:H7,7,FALSE)&" years at an interest rate of "&VLOOKUP(Instructions!B2,Information!A2:H7,8,FALSE)&"%."
Label in Cell A4: =CONCATENATE(VLOOKUP(Instructions!B2,Information!A2:H7,5,FALSE)," Value")
Label in Cell C8: =IF(VLOOKUP(Instructions!B2,Information!A2:H7,5,FALSE)="Present","FV","PV")&" (formula)"
Label in Cell D8: =IF(VLOOKUP(Instructions!B2,Information!A2:H7,5,FALSE)="Present","FV","PV")&" (function)"