VBA: Tutorial Week 3

- A3 requirements & tips for success
- Non-linear, non-sequential programming using branches

Official resource for MS-Office products: https://support.office.com

First Tutorial (Monday or Tuesday): A3

CPSC 203: A3

- This is your first 'real' program writing assignment so you should not under estimate the time and effort required by typical students.
- Here's some "tips for success" that are given to students who are first learning how to write programs (exert from CPSC 217 and CPSC 231)
  - Practice things yourself
  - Make sure that you keep up with the material
    - Study concepts as soon as possible after class
    - If you fall behind more than 1 or 2 weeks without studying then you may not be able to catch up! ☹
  - Start working on things early

VBA tutorial notes by James Tam

A3: Feature #1

- Prompt the user for a word to find in a document(s).
- If Feature #6 is implemented the same word will be 'found' by the program in each document opened by your program.

VBA tutorial notes by James Tam

A3: Feature #2

- Count the number of instances of the word (from Feature #1) using the appropriate parts of the ActiveDocument object.
- Not case sensitive.
- Partial matches are counted.

A3: Feature #3

- Display the count of the instances via a MsgBox in the following form. "Number occurrences: <*actual number of occurrences of the search word*>"
  - e.g. Number occurrences: 2

A3: Feature #4

- Write information about the number of instances into the currently active Word document using the appropriate parts of the Selection object.
- (If Feature #6 is implemented then this information would be written into the respective document in which the count was being conducted).

VBA tutorial notes by James Tam

A3: Feature #4 (2)

- Format of the written text:
  a) The text written should take the form: "Number occurrences: <*actual number of occurrences of the search word*>"
     - e.g. Number occurrences: 2
  b) This text should appear on its own line i.e. it should be preceded by "a hard return" or a VBA carriage return "vbCr".
  c) The text appears at the end of the document ("end of story").
  d) If the search word (entered with Feature #1) appears two times or more then the text (from Feature #4a) to be written (Feature 4a) is colored red. Credit will be awarded if the text is colored under the correct condition.
  e) If the search word appears three or more times then the text to be written (from Feature #4a) is also bolded. Credit will be awarded if the text is bolded under the correct condition.

VBA tutorial notes by James Tam

A3: Feature #5

- Double the size (both the height and width change so the proportions of height to size remains the same) of each image (in VBA "InLineShapes" in the document.
- If the document doesn't contain any images then a MsgBox will appear with the message "No images to modify".

A3: #Feature #6

- Program runs Features 2 - 5 on **all the Word documents** at a location specified by the user.
- Features #2-5 are contained in the body of a loop defined for Feature #6.
  - Feature #1 runs only once, the program searches for the same word for **all** Word documents.
-

A3: Parts Of Feature #6

- Prompt the user for the location of the Word documents using an InputBox e.g. e.g. "C:\203".
  - The program should not require that the user enters a slash after the name of the last folder (in this example the containing folder is '203') because your VBA program will automatically add it to the end before trying to open any documents.
- If the user enters an empty location the *program* will display a MsgBox that contains the message "No location entered, ending program" and then the program will end.
- If the location is not empty then the program will successively open each Word document the at that location using the appropriate parts of the Documents collection.

VBA tutorial notes by James Tam

A3: Parts Of Feature #6 (2)

- The program must **not only open just Word documents** but only open certain types of Word documents:
  - older **Word 97-2003 (.doc)** documents,
  - newer **Word 2007+ (.docx)** documents,
  - **macro enabled documents (.docm)**.
- After opening each Word document the program will apply Features 2 - 5 on each document.

VBA tutorial notes by James Tam

A3: Documentation Requirements

- Contact information: your full name as provided to the university (make sure it matches, don't include 'nicknames'), student identification number, tutorial number.
- Specifying clearly what features of the assignment that you completed or didn't complete. In order to get credit the documentation has to be clear and complete.

A3: Style Requirements

- Each level of code indenting is consistently 1 tab.
  - Instructions in the sub-routine (between the 'sub' and 'end-sub' is 1 level.
  - The body of IF or WHILE structures counts as another level of indenting.
  - A tab is used for each level of indenting
- Example:

```
Sub exampleIndenting()
    Do while()
        If () then
             if () then
             End if
        else
        End if
    Loop
End sub
```

A3: Style Requirements (2)

- Good naming conventions (e.g. variables, sub-routines, the name of Word document containing the VBA program and constants if applicable) are followed. Some examples of naming conventions are provided in [the VBA Part I notes].
  - https://pages.cpsc.ucalgary.ca/~tamj/2022/203W/notes/pdf/vba_part1.pdf

VBA tutorial notes by James Tam

Microsoft Introduction/Overview Of VBA

- https://docs.microsoft.com/en-us/office/vba/library-reference/concepts/getting-started-with-vba-in-office

VBA tutorial notes by James Tam

Second Tutorial (Wednesday or Thursday)

Activities In Tutorial

- TA demos:
  - Used for more complex features (typically multiple steps are required).
  - The tutorial instructor will show on the projector/instructor computer each step for running the feature in Excel.
  - Unless otherwise specified the tutorial material will take the form of a TA demonstrating the use of features in Excel.
  - Slides titled “Lecture Review” are covered for the second time and dealing with less complex material.
    - For this reason they will only be covered briefly in tutorial.
- Student exercises:
  - Used instead of TA demos for simpler features.
  - You will have already been given a summary of how to invoke the feature and the purpose of the exercise is to give you a chance to try it out and get help if needed.

VBA tutorial notes by James Tam

Student Exercise #1

- (You may need to review the lecture notes on collections before trying this exercise)
- Using the starting document called "`1sorting_table_starting`" write a VBA program that will sort the first table (ascending order, assumes there is a header) if the document contains more than one table.
- If the cut-off for the number of tables hasn't been met then the program should display a popup message box with a brief description of why sorting didn't occur as well as the number of tables in the document.
- (JT's comment: test your program by seeing what happens if it contains: no tables, 1 table, 2 or more tables).
- Document containing the solution: `1sorting_table_solution`

VBA tutorial notes by James Tam

Branching: Alternate Courses Of Execution

- What you will know from lecture:
  - Branching allows for alternative courses of execution.
  - Each alternative executes one or more VBA instructions.
  - Branching can be implemented in different ways depending upon the programming language what you will have learned is variations of the `IF` structure.

VBA tutorial notes by James Tam

IF-THEN ; IF-THEN, ELSE

- A Boolean expression (results in either a true or false value) will determine the instruction or instructions that will execute.
- IF-THEN executes an instruction or instructions (body of the IF) when the Boolean expression evaluates to true.
  - E.g.

    ```
    If (true) Then
    ```

    - The above examples always executes the body because the Boolean expression is always true (the expression is a constant value that is true).
- IF-THEN, ELSE reacts for both the true and the false cases.
  - E.g.

    ```
    If (num >= 0) Then
    Else
    ```

    - Executes the IF-body when num is positive and the Else-body when num is not positive.

Branches: Depending Upon The # Of Images

- **Name of the document containing example**: 1ifElse
- Features: checks if the number of images ('InlineShapes' in VBA) is above the defined cut-off value).
  - ifThenExample: The first program reacts (popup appears) if the cut-off has been met.
  - ifThenElseExample: The second program reacts one way if the cut-off has been met (popup appears) and another way (different popup appears)
  - After the branch has been completed both programs will then execute any remaining instructions after the branching structure (after the 'End If')
  - Also:
    - shows an example of defining a named constant (specifies the cut-off value),
    - shows the use of the line continuation character (string argument for the MsgBox in the body of the ELSE-branch in the second program).

Branches: Depending Upon The # Of Images IF-Then Version

```
' First program (IF-THEN)
Sub ifThenExample()
    Const CUT_OFF As Long = 2
    Dim numShapes As Long
    numShapes = ActiveDocument.InlineShapes.Count
    If (numShapes > CUT_OFF) Then
        MsgBox (">" & CUT_OFF & " pics in active Word doc")
    End If
    MsgBox ("Branching structure over: End program")
End Sub
```


Branches: Depending Upon The # Of Images IF-Then, Else Version

```
' Second program (IF-THEN, ELSE)
Sub ifThenElseExample()
    Const CUT_OFF As Long = 2
    Dim numShapes As Long
    numShapes = ActiveDocument.InlineShapes.Count
    If (numShapes > CUT_OFF) Then
        MsgBox (">" & CUT_OFF & " pics in active Word doc")
    Else
        MsgBox ("# pics didn't meet the cutoff of " & CUT_OFF & _
         " pics " & " required ")
    End If
    MsgBox ("Branching structure over: End program")
End Sub
```

Logic And Branching

- Recall how the logical functions, AND() OR(), can be combined with an IF-function in Excel.
  - E.g. (hires U of C graduates with a GPA of 3.3 or higher)
    `IF(AND(A1>3.3,B1="UC", "Hire", "")`
- With programming languages the structure is slightly different, Boolean expressions are chained or connected with AND OR logical operators.
  - **Format**:
    - `If ((Boolean expression 1) <Logical operator> (Boolean expression 2)…) then`
  - **Example**:
    - `If ((age < 0) Or (age > 114)) Then`
  - `(More than 2 Boolean expressions can be evaluated by the use of additional logical operators):`
    - `If ((age < 0) Or (age > 114) Or (hair = "mullet")) Then`

VBA tutorial notes by James Tam

Logic & Branching

- **Name of the document containing example**: `2branchingLogic`
- Features: Error checks user input using branching (both branches produce similar results)
  - First branch example: Employs logical OR (checks if it's true age is outside the valid range).
    - Shows an error message if it's true that the age is outside the allowable range.
    - Shows a confirmation message if it's false that the age is outside the allowable range (i.e. the age is okay)
  - Second branch example: Employs logical AND (checks if it's true age is inside the valid range)
    - Shows a confirmation message if it's true that the age is inside the allowable range
    - Shows an error message if it's false that the age is inside the allowable range (i.e. the age is not okay)
  - Both branches produce the same popup given the same user input (student: verify this for practice).

VBA tutorial notes by James Tam

Logic & Branching: Error Checking Age

- VBA instructions needed for both branches

```
Dim age As Long
age = InputBox("Age (0 - 114)?")
```


Error Checking A Value: IF With OR

```
If ((age < 0) Or (age > 114)) Then
    MsgBox ("OR: Age is outside allowable range of 0 - 114")
Else
    MsgBox ("OR: Age of " & age & " is OK")
End If
```

Error Checking A Value: IF With AND

```
If ((age >= 0) And (age <= 114)) Then
    MsgBox ("AND: Age of " & age & " is OK")
Else
    MsgBox ("AND: Age is outside allowable 0 - 114")
End If
```


Checking Multiple Conditions

- There's two general cases:
  - Zero or one of the conditions is true (no more than one so having one true case excludes the possibility of any other cases being true).
    - Example: getting a letter grade for a class during a particular semester, specifying the current city, town that you reside in.
    - VBA structure to use: IF-ELSEIF
  - Zero, one, two up to all of the cases can be true.
    - Example: for each class taken checking if a perfect score was awarded (letter grade 'A'), checking if a person has ever lived in each of the cities, towns in a particular country (Have you ever lived in Calgary? Have you ever lived in Edmonton? Etc.)
    - VBA structure to use: Multiple and independent IFs

IF-ELSEIF, Multiple IFs

- **Name of the document containing example**: `3multipleIfVSIFElseIF`
- Features:
  - Part I: Check for birth city
    - Prompts the user for the city that person was born in an reacts in different ways based on that information. One approach uses an `IF-ELSEIF` structure, the other employs multiple and separate `IF` structures.
    - Note: the use of the multiple IFs is not an appropriate solution in cases such as this but is included for learning purposes (to show how difficult it can be to check for the 'none of the above' case).
  - Part II: Check education level and if the person is a senior citizen
    - These two checks are independent, education level and the senior check are unrelated.
    - (Alternatively phrased): regardless of what the user enters for the years of education, the program will always check if the person is or is not a senior.

VBA tutorial notes by James Tam

Multiple Conditions: Checking City Of Birth

- Solution using multiple `IF`s

```
Dim birthCity As String
birthCity = InputBox("City of birth")
If (birthCity = "Calgary") Then
    MsgBox ("You are 'Part of the Energy'")
End If
If (birthCity = "Edmonton") Then
    MsgBox ("From the City of Champions")
End If
If (birthCity = "Dubai") Then
    MsgBox ("Definitely Dubai!")
End If
If (birthCity = "Fargo") Then
    MsgBox ("You're always warm")
End If
```

VBA tutorial notes by James Tam

Multiple Conditions: Checking City Of Birth (2)

```
' When none of the cases in any of the previous IFs apply
' requires a very awkward solution
If ((birthCity <> "Calgary") And _
    (birthCity <> "Edmonton") And _
    (birthCity <> "Dubai") And _
    (birthCity <> "Fargo")) Then
      MsgBox ("Multiple-IFs: Miscellaneous place")
End If
```


Multiple Conditions: Checking City Of Birth (3)

- Solution using `IF-ELSEIF` (better approach)

```
If (birthCity = "Calgary") Then
    MsgBox ("You are 'Part of the Energy'")
ElseIf (birthCity = "Edmonton") Then
    MsgBox ("From the City of Champions")
ElseIf (birthCity = "Dubai") Then
    MsgBox ("Definitely Dubai!")
ElseIf (birthCity = "Fargo") Then
    MsgBox ("You're always warm")
Else
    MsgBox ("IF-ELSEIFs: Miscellaneous place")
End If
```

Multiple Conditions: Education Level, Senior Citizen (Solution Has A Bug)

- Student exercise #2: find the bug in the following program.

```
Dim gradeLevel As Long
Dim age As Long
gradeLevel = InputBox("What is your highest grade level: ")
age = InputBox("What is your age: ")
If (gradeLevel >= 13) Then
    MsgBox ("College person!")
ElseIf (age >= 65) Then
    MsgBox ("Senior citizen")
End If
```

VBA tutorial notes by James Tam

Multiple Conditions: Grading Program (Solution Has A Bug)

- Student exercise #3: find the bug in the following program.

```
Dim gradePoint As Double
gradePoint = InputBox("Type in the grade point: ")
If (gradePoint == 0) Then
    MsgBox ("Failing grade")
ElseIf (gradePoint > 0) Then
    MsgBox ("Minimal pass")
ElseIf (gradePoint > 1) Then
    MsgBox ("Satisfactory")
ElseIf (gradePoint > 2) Then
    MsgBox ("Excellent")
ElseIf (gradePoint > 3) Then
    MsgBox ("Perfect")
End If
```

Grading scales
- GPA over 3.0 is 'Perfect'
- GPA over 2.0 is 'Excellent
- GPA over 1.0 is 'Satisfactory'
- GPA over 0.0 is a passing grade
- GPA of 0.0 is a failing grade

VBA tutorial notes by James Tam

Second Tutorial (Wednesday or Thursday)

Open Tutorial

- No new teaching will occur but the TA will be available for help. During this "Open Tutorial"
- Any CPSC 203 student can ask for help and not just the students who are registered in a particular tutorial.
- The purpose is to provide extra help because the next workbook exercise is the first one in which you need to write a program from scratch.

VBA tutorial notes by James Tam