VBA: Tutorial Week 6

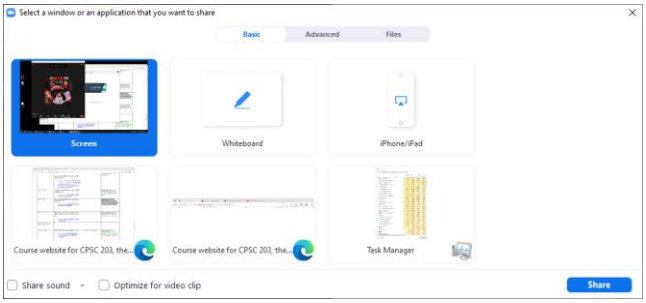
- Going over the Assignment 4 requirements.
- Formatting cells: setting the fill color, changing fonts and font effects
- Accessing cell data
- Inserting and simple configuring of chart properties
- Accessing specific worksheets in the currently active workbook
- Data analysis: counting occurrences, specifying search criteria
- Sorting spreadsheets
- A return to nested loops

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

First Tutorial: Monday Or Tuesday

FYI For The Tutorial Instructor/TA

- Since you will be playing a video with a voice narration make sure that you have enabled the “Share sound” option if you are using Zoom.
  - You might want to use the pulldown (triangle) to ensure that audio is set to ‘stereo’ rather than ‘mono’.



VBA tutorial notes by James Tam

VBA Programming For Excel

- Many of the programming tools or structures you have been taught during the VBA-Word lectures and tutorials are applicable (e.g. documentation, variables, constants, getting input via a MsgBox, showing output messages via an InputBox, branching, loops etc.).
- For this course: only a few things will be new such as Excel objects, methods/functions.
- This is why so much time was provided to complete A3 but far less time is provided for A4.
  - For A3 you were learning many new tools for the first time.
  - For A4 you are applying previous tools and only a few new ones.

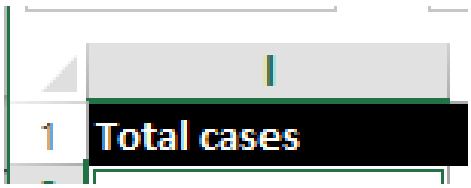
VBA tutorial notes by James Tam

A4: Overview Video

- This assignment allows you to work on a real world problem.
  - Tracking actual Covid infection data in Alberta.
- (To the tutorial instructor: in the interests of time you may want to show only select parts).
  - https://www.youtube.com/watch?v=A0GuUGfvI5c&feature=youtu.be

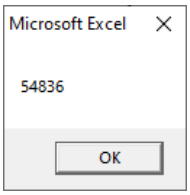
A4: Feature #1

- Write the text "Total cases" into the cell `I1`.



A4: Feature #2

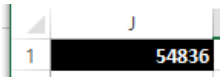
- Counts the total number of Covid cases and displays this information in a MsgBox.
  - It includes all cases status such as 'Recovered', 'Active' etc.
  - The count must be conducted with a loop and variable is used to track the count.



VBA tutorial notes by James Tam

A4: Feature #3

- Writes the count from the previous feature (i.e. the MsgBox output) to Cell J1.



- Obviously what you write in the cell to the immediate left augments the number written into this cell.

VBA tutorial notes by James Tam

A4: Feature #4

- The two Cells (`I1` and `J1`) where the information written from the previous two features are bolded.

A4: Feature #5

- The data in the spreadsheet is sorted by date (earliest to latest).

A4: Feature #6

- The date in which infection data appears in the spreadsheet (i.e. the entry of a date in the rows of Column B signify that an infection occurred on that date) will be written into Column `I`.
- Regardless of the number of infections that occurred on a particular day, the date information is only written into Column `I` once.

	B	C	D	E	F	G	H	I	
2	**Date reported**	**Alberta Health Services Zone**	**Gender**	**Age group**	**Case status**	**Case type**		**Date**	**New**
3	2020-03-06	Calgary Zone	Female	50-59 years	Recovered	Confirmed		2020-03-06	1
4	2020-03-09	Calgary Zone	Female	30-39 years	Recovered	Confirmed		2020-03-09	6
5	2020-03-09	Calgary Zone	Male	30-39 years	Recovered	Confirmed		2020-03-10	9
6	2020-03-09	Edmonton Zone	Male	60-69 years	Recovered	Confirmed		2020-03-11	9
7	2020-03-09	Edmonton Zone	Male	40-49 years	Recovered	Confirmed		2020-03-12	2
8	2020-03-09	Edmonton Zone	Female	70-79 years	Recovered	Confirmed		2020-03-13	8
9	2020-03-09	Calgary Zone	Female	50-59 years	Recovered	Confirmed		2020-03-14	19
10	2020-03-10	Edmonton Zone	Male	70-79 years	Recovered	Confirmed		2020-03-15	9

VBA tutorial notes by James Tam

A4: Feature #7

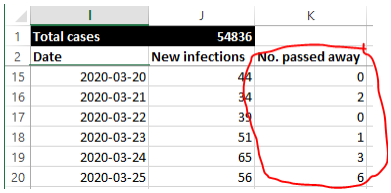
- (Nesting is mandatory)
  - You will have an outer loop.
  - The body of the outer loop contains a nested inner loop or a nested branching structure.
- The number of new infections for a particular date are written into the rows of Column `J` with the first date appearing at Cell `J3`.
- Because the date only appears once in Column `I` the number of infections for a particular day will appear only once on one row.

	I	J	K
1	**Total cases**	**627**	
2	**Date**	**New infections**	**Number who passed away**
3	2020-03-06	1	0
4	2020-03-09	6	0
5	2020-03-10	9	0

VBA tutorial notes by James Tam

A4: Feature #8

- (As with the previous feature nesting is mandatory).
- The number of people who passed away on particular date will be written into the rows of Column K.
- Similar to new infections only the total number of people who passed away on a particular date will be written and written once.



	I	J	K
1	Total cases	54836	
2	Date	New infections	No. passed away
15	2020-03-20	44	0
16	2020-03-21	34	2
17	2020-03-22	39	0
18	2020-03-23	51	1
19	2020-03-24	65	3
20	2020-03-25	56	6

VBA tutorial notes by James Tam

A4: Feature #9

- (To get credit for this feature the previous 3 features need to correctly and completely implemented).
- Insert a chart that graphs: the date, number of infections for each date, the number who passed away on each date.
- Whether one chart is used for new infections and deaths, or one chart graphs infections over time while the other graphs deaths over time the same credit will be awarded.
- Acceptable charts include: line, bar or column.
- It's best to stick to one of the 3 specified types and use a 2D graph only.

VBA tutorial notes by James Tam

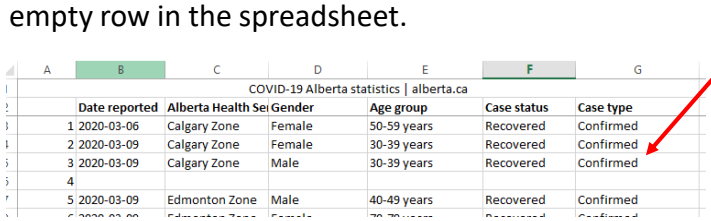
A4: Feature #10

- (Requires the previous feature to be complete and correct). The chart title must be changed to include the text "Alberta Covid statistics".
- If you have a separate chart for new infections and deaths then each chart can include this text plus more specific information e.g. "Alberta Covid statistics: New infections" and "Alberta Covid statistics: Number of people who passed away".

VBA tutorial notes by James Tam

A4: Where Does It All End?

- The end of the infection data will always be followed by an empty row in the spreadsheet.



	A	B	C	D	E	F	G
		COVID-19 Alberta statistics | alberta.ca					
		Date reported	Alberta Health Se	Gender	Age group	Case status	Case type
	1	2020-03-06	Calgary Zone	Female	50-59 years	Recovered	Confirmed
	2	2020-03-09	Calgary Zone	Female	30-39 years	Recovered	Confirmed
	3	2020-03-09	Calgary Zone	Male	30-39 years	Recovered	Confirmed
	4						
	5	2020-03-09	Edmonton Zone	Male	40-49 years	Recovered	Confirmed

- (As shown in lecture on March 31, reducing the number of cases can make it easier to trace/debug a program).

VBA tutorial notes by James Tam

A4: Documentation Requirements

- Contact information: your full name, student identification number, tutorial number (**New for A4)**
- **New for A4:** Demonstrate some evidence of a versioning system. The program that you submit must specify at least one version number (a date is acceptable).
- If your program includes more than one version then list the features completed for each version.
- More information about versioning (and documentation) is provided in [the VBA Part I notes and lectures]

VBA tutorial notes by James Tam

A4: Style Requirements

- Each level of code indenting is consistently 1 tab.
- Good naming conventions (e.g. variables, sub-routines, the name of Word document containing the VBA program and constants if applicable) are followed.
- **New for A4:** The use of named constants (One of many possible examples: Const EMPTY_CELL As String = "") as appropriate.
  - Named constants were first introduced in the first VBA programming set of notes. Examples have been shown in some of the subsequent Word examples and many were shown in the two "VBA Extras" lectures.

VBA tutorial notes by James Tam

Example: Changing Fonts, Font Effects, Fill Color

- Font changes can be made via the Cells or the Range object
- **Spreadsheet name**: 1_formatting_cells

```
Sub formattingEffects()
    Dim colorChoice As String
    Dim colorChoiceInvalid As Boolean
    Range("B2:D5").Font.Name = "Arial Black"
    Cells(1, 1).Font.Size = 24
    colorChoiceInvalid = True
```

VBA tutorial notes by James Tam

Example: Changing Fonts, Font Effects, Fill Color (2)

```
    Do While (colorChoiceInvalid = True)
        colorChoiceInvalid = False
        colorChoice = InputBox("Color (red,blue,green): ")
        If ((colorChoice <> "red") And _
            (colorChoice <> "blue") And _
            (colorChoice <> "green")) Then
            colorChoiceInvalid = True
        ElseIf (colorChoice = "red") Then
            Range("C3:E7").Interior.Color = vbRed
            Range("C3:E7").Font.Color = vbWhite
            Range("C3:E7").Font.Bold = True
        ElseIf (colorChoice = "blue") Then
            Range("C3:E7").Interior.Color = vbBlue
            Range("C3:E7").Font.Color = vbYellow
            Range("C3:E7").Font.Bold = True
```

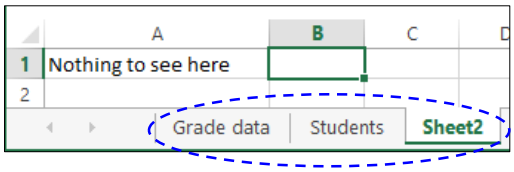
VBA tutorial notes by James Tam

Example: Changing Fonts, Font Effects, Fill Color (3)

```
        ElseIf (colorChoice = "green") Then
            Range("C3:E7").Interior.Color = vbGreen
            Range("C3:E7").Font.Color = vbBlue
        End If
    Loop
End Sub
```


Accessing Worksheets

- Much like with a VBA program where instructions typically affect the currently active document, programs written for Excel will affect the *currently active worksheet*.
- Worksheets can either be accessed by the **name** or the *order in which the sheet was added* to the spreadsheet (not the left-right ordering).



Example: Accessing Specific Worksheets

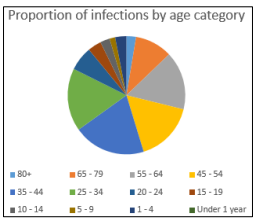
- **Spreadsheet name**: `2_accessing_worksheets_by_user_input`

```
Sub accessingWorksheets()
    Dim worksheetName As String
    Dim worksheetNumber As Long
    worksheetName = InputBox("Worksheet name to change (Grade
      data, Students, Sheet2): ")
    worksheetNumber = InputBox("Worksheet number to change (1-
      3): ")
    Worksheets(worksheetName).Range("A1") = "Made change to
      worksheet " & worksheetName
    Worksheets(worksheetNumber).Range("B1") = "Made change to
      worksheet #" & worksheetNumber
End Sub
```

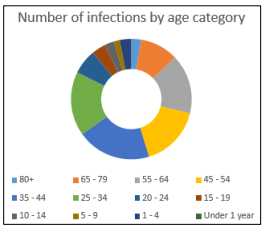
VBA tutorial notes by James Tam

Commonly Used Charts To Represent Proportions

- Pie chart



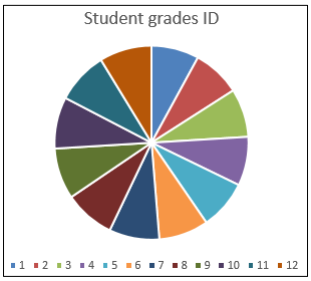
- Donut chart

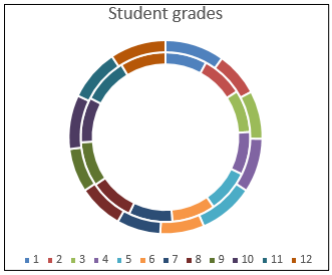


VBA tutorial notes by James Tam

Pie And Donut Charts: When Not To Use

- These types of representations are poor at representing exact numeric values (e.g. what was the grade for student #6?).
  - Yet they are sometimes used this way in real life!





VBA tutorial notes by James Tam

Example: Inserting Charts Representing Proportions

- **Spreadsheet name**: `3_inserting_portional_charts`

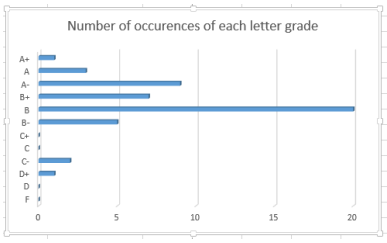
```
Sub insertPieChart()
    Range("A2:A14,C2:D14").Select
    ActiveSheet.Shapes.AddChart2(201, xlPie).Select
    ActiveChart.ChartTitle.Select
    ActiveChart.ChartTitle.Text = "Proportion of infections by age"
End Sub

Sub insertDonutChart()
    Range("A2:A14,C2:C14").Select
    ActiveSheet.Shapes.AddChart2(201, xlDoughnut).Select
    ActiveChart.ChartTitle.Select
    ActiveChart.ChartTitle.Text = "Number of infections by age"
End Sub
```

VBA tutorial notes by James Tam

Example: Inserting Charts Representing Quantities

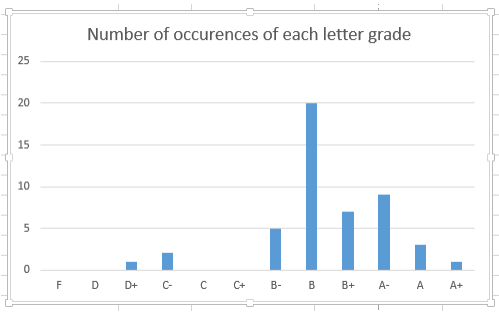
- Some good choices include bar, column and line charts
- **Spreadsheet name**: 4_inserting_quantitative_charts
  - Bar chart

```
Sub insertBarChart()
    Range("C2:D13").Select
    ActiveSheet.Shapes.AddChart2(201, xl3DBarClustered).Select
    ActiveChart.ChartTitle.Select
    ActiveChart.ChartTitle.Text = "Number of occurrences"
End Sub
```



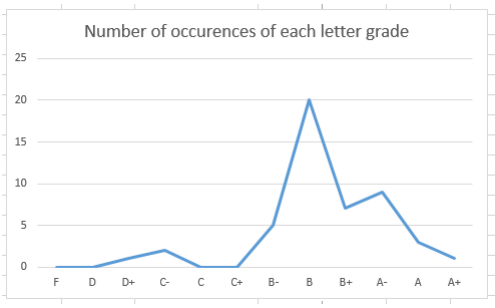
Example: Inserting Charts Representing Quantities (2)

```
Sub insertColumnChart()
    Range("C2:D13").Select
    ActiveSheet.Shapes.AddChart2(201, xlColumnClustered).Select
    ActiveChart.ChartTitle.Select
    ActiveChart.ChartTitle.Text = "Number of occurrences"
End Sub
```



Example: Inserting Charts Representing Quantities (3)

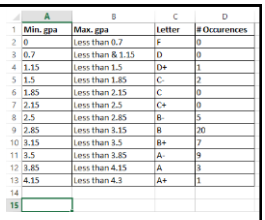
```
Sub insertLineChart()
    Range("C2:D13").Select
    ActiveSheet.Shapes.AddChart2(201, xlLine).Select
    ActiveChart.ChartTitle.Select
    ActiveChart.ChartTitle.Text = "Number of occurrences"
End Sub
```



VBA tutorial notes by James Tam

Second Tutorial: Wednesday Or Thursday

Counting The Number Of Rows In A Chart



	A	B	C	D
1	Min. gpa	Max. gpa	Letter	# Occurences
2	0	Less than 0.7	F	0
3	0.7	Less than & 1.15	D	0
4	1.15	Less than 1.5	D+	1
5	1.5	Less than 1.85	C-	2
6	1.85	Less than 2.15	C	0
7	2.15	Less than 2.5	C+	0
8	2.5	Less than 2.85	B-	5
9	2.85	Less than 3.15	B	20
10	3.15	Less than 3.5	B+	7
11	3.5	Less than 3.85	A-	9
12	3.85	Less than 4.15	A	3
13	4.15	Less than 4.3	A+	1
14				
15				

- With a small set of data you may be able to do this.
  - What if you wanted to do this for many spreadsheets (instances of non-empty rows in 1000+ sheets).
- A loop can be used to step through row by row until an empty row has been encountered.

VBA tutorial notes by James Tam

Example: Counting Rows

- **Name of spreadsheet:** `5_counting_rows_for_chart`

```
This program will only include in the chart the actual number
of rows of data.

Sub countingRowsToShart()
    Const LETTER_GRADE_COLUMN As Long = 3
    Const START_ROW As Long = 1
    Const EMPTY_ROW As String = ""
    Dim rowData As String
    Dim currentRow As Long
    Dim count As Long
```

VBA tutorial notes by James Tam

Example: Counting Rows (2)

```
    'Counting number of rows
    currentRow = START_ROW
    count = 0
    rowData = Cells(currentRow, LETTER_GRADE_COLUMN)
    Do While (rowData <> EMPTY_ROW)
        count = count + 1
        currentRow = currentRow + 1
        rowData = Cells(currentRow, LETTER_GRADE_COLUMN)
    Loop

    'Insert chart based on range. Will always start at C1
    'but last row determined by number of rows
    Range("C1" & ":" & "D" & count).Select
    ActiveSheet.Shapes.AddChart2(201, xlLineMarkers).Select
End Sub
```


Exercise 1

- **Program description:**
  - Counts the number of rows containing data (headings and student data).
  - The count will be written to cell address that is specified by the user.
- **Spreadsheet containing the solution** (don't look at it until you have at least made an attempt):
  `Exercise1_counting_rows_writing_user_specified_location`

Example: Counting Instances Of User Specified Search Criteria

- **Spreadsheet name**:
  6_searching_spreadsheets_with_user_critiera_writing_results

```
Sub searchV1()
    Const EMPTY_DATA As String = ""
    Const MEMBERS_ROW As Long = 2
    Const START_RESULTS_ROW As Long = 17
    Const SEARCH_CRITERIA_COLUMN = 2
    Const MEMBER_COLUMN As Long = 1
    Const ETHNICITY_COLUMN As Long = 2
    Const CITY_COLUMN As Long = 3
    Const AGE_COLUMN As Long = 4
    Const NUMBER_MATCHES_ROW As Long = 15
    Const NUMBER_MATCHES_COLUMN As Long = 2
```


Example: Counting Instances Of User Specified Search Criteria (2)

```
    Dim count As Long
    Dim searchRow As Long
    Dim currentResultsRow As Long
    Dim desiredCity As String
    Dim currentMemberName As String
    Dim minAge As Long
    Dim maxAge As Long
    Dim currentMemberCity As String
    Dim currentMemberAge As Long

    desiredCity = InputBox("City: ")
    minAge = InputBox("Youngest age for search: ")
    maxAge = InputBox("Oldest age for search: ")
```

Example: Counting Instances Of User Specified Search Criteria (3)

```
Do While (currentMemberName <> EMPTY_DATA)
    currentMemberCity = Cells(searchRow, CITY_COLUMN)
    currentMemberAge = Cells(searchRow, AGE_COLUMN)
    If ((desiredCity = currentMemberCity) And _
           ((currentMemberAge >= minAge) And _
            (currentMemberAge <= maxAge))) Then
           count = count + 1
        Cells(currentResultsRow, MEMBER_COLUMN) = Cells(searchRow,
          MEMBER_COLUMN)
        Cells(currentResultsRow, SEARCH_CRITERIA_COLUMN) =
          desiredCity & ", " & currentMemberAge
         currentResultsRow = currentResultsRow + 1
       End If
       searchRow = searchRow + 1
       currentMemberName = Cells(searchRow, MEMBER_COLUMN)
   Loop
```

VBA tutorial notes by James Tam

Example: Counting Instances Of User Specified Search Criteria (4)

```
   'Write out total number of matches
   Cells(NUMBER_MATCHES_ROW, NUMBER_MATCHES_COLUMN) = count
End Sub
```

VBA tutorial notes by James Tam

Exercise 2

- **Program description:**
  - Counts the number of occurrences of a Covid status (e.g. Recovered, Died) in the spreadsheet.
  - The status is entered by the user.
  - The count will be written to row 3, column 10 (Cell J3).
- **Spreadsheet containing the solution** (don't look at it until you have at least made an attempt):
  `Exercise2_covid_data_counting_number_of_user_selected_occurance`

VBA tutorial notes by James Tam

Exercise 3

- **Program description:**
  - Starting with the solution to the previous exercise modify the program so the user can also select the location where results are written to the spreadsheet.
  - It's your choice if the destination is determined by a (row, column) integer pair or through a cell address.

Spreadsheet containing the solution (don't look at it until you have at least made an attempt):
`Exercise3_covid_data_user_selects_start_and_end_count_range_and output location`

VBA tutorial notes by James Tam

Example: Error Checking Input, Sorting Based On User Criteria

- **Name of spreadsheet**:
  7_error_checking_input_sorting_by_user_criteria

```
Sub errorCheckingSortingGrades()
    Dim sortCriteria As String
    Dim sortKey As String

    sortCriteria = InputBox("Sort criteria: ID', 'Last Name', 'GPA'")
    Do While ((sortCriteria <> "ID") And _
              (sortCriteria <> "Last Name") And _
              (sortCriteria <> "GPA"))
        sortCriteria = InputBox("Sort criteria: ID', 'Last name',
          'GPA'")
    Loop
```


Example: Error Checking Input, Sorting Based On User Criteria (2)

```
    If (sortCriteria = "ID") Then
        sortKey = "A1"
    ElseIf (sortCriteria = "Last Name") Then
        sortKey = "B1"
    ElseIf (sortCriteria = "GPA") Then
        sortKey = "E1"
    End If

    ActiveWorkbook.Worksheets(1).Sort.SortFields.Clear
    ActiveWorkbook.Worksheets(1).Sort.SortFields.Add Key:= _
        Range(sortKey), Order:=xlAscending
    With ActiveWorkbook.Worksheets(1).Sort
        .SetRange Range("A1:F12")
        .Header = xlYes 'Options: x1No, x1yes
        .Apply
    End With
End Sub
```

Note: there is a recorded macro you can see in the VB editor that shows how to sort by multiple keys

Nesting

- Two repeated processes.
  - One is nested inside the other.
  - That means that each time one process starts the nested/inner process starts from beginning to end.
- Examples of **nested loops** (non-exhaustive list) from lecture.
  - **Washing Dishes**

```
While (there are dishes left unwashed)
    Get a dirty dish
    Apply soap to dish
      while (dish is still dirty)
       Rub dish with wet cleaning tool
       If (more soap needed)
          Apply soap to dish
```

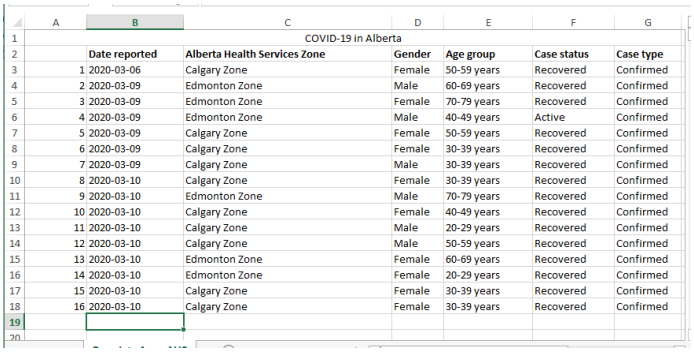

Nesting (2)

- Examples of **nested loops** (non-exhaustive list) from lecture (continued):
  - **Martial Arts**

```
While (there is still a compass point with opponent)
    Turn left to face opponent
      while (opponent is still standing)
       Throw right reverse punch
       Left rising block
       Throw right reverse punch
     Assume guard position
```

Nesting (3)

- Examples of nested (non-exhaustive list) from lecture (continued):
  - **Counting Covid Alberta Cases**



	A	B	C	D	E	F	G
1		COVID-19 in Alberta					
2		Date reported	Alberta Health Services Zone	Gender	Age group	Case status	Case type
3	1	2020-03-06	Calgary Zone	Female	50-59 years	Recovered	Confirmed
4	2	2020-03-09	Edmonton Zone	Male	60-69 years	Recovered	Confirmed
5	3	2020-03-09	Edmonton Zone	Female	70-79 years	Recovered	Confirmed
6	4	2020-03-09	Edmonton Zone	Male	40-49 years	Active	Confirmed
7	5	2020-03-09	Calgary Zone	Female	50-59 years	Recovered	Confirmed
8	6	2020-03-09	Calgary Zone	Female	30-39 years	Recovered	Confirmed
9	7	2020-03-09	Calgary Zone	Male	30-39 years	Recovered	Confirmed
10	8	2020-03-10	Calgary Zone	Female	30-39 years	Recovered	Confirmed
11	9	2020-03-10	Edmonton Zone	Male	70-79 years	Recovered	Confirmed
12	10	2020-03-10	Calgary Zone	Female	40-49 years	Recovered	Confirmed
13	11	2020-03-10	Calgary Zone	Male	20-29 years	Recovered	Confirmed
14	12	2020-03-10	Calgary Zone	Male	50-59 years	Recovered	Confirmed
15	13	2020-03-10	Edmonton Zone	Female	60-69 years	Recovered	Confirmed
16	14	2020-03-10	Edmonton Zone	Female	20-29 years	Recovered	Confirmed
17	15	2020-03-10	Calgary Zone	Female	30-39 years	Recovered	Confirmed
18	16	2020-03-10	Calgary Zone	Female	30-39 years	Recovered	Confirmed
19							
20							

VBA tutorial notes by James Tam

Nesting (4)

- Another example: Workbook exercise

	A	B	C	D
1	Month	Day	Number of visitors	Using sum to count visitors/month
2	Jan	2	15	139
3	Jan	12	25	
4	Jan	24	38	
5	Jan	31	61	
6	Feb	13	52	92
7	Feb	14	40	
8	Mar	1	100	319
9	Mar	14	98	
10	Mar	25	53	
11	Mar	30	68	
12	Apr	1	27	147
13	Apr	9	100	
14	Apr	10	20	
15	Jun	2	67	241
16	Jun	3	43	
17	Jun	4	54	
18	Jun	5	77	
19	Jul	1	69	142
20	Jul	30	73	

VBA tutorial notes by James Tam

Example: Nesting

- This program prompts the user for a North American country.
- It will re-prompt so long as the country name isn't one of three possibilities.
- Each time the user enters a valid country the program will check if valid region has been entered (currently program only cross checks Canada with Canadian provinces).
- Again the program re-prompts for a region until a valid one has been entered.
- **Spreadsheet name**: `8_nested_loops_country_city_count`

Example: Nesting (2)

```
Sub countClients()
    Const COUNTRY_COLUMN As Long = 1
    Const REGION_COLUMN As Long = 2
    Const NO_VALUE As String = ""
    Const START_ROW As Long = 3
    Dim country As String
    Dim region As String
    Dim countryCount As Long
    Dim regionCount As Long
    Dim row As Long
    Dim countryFromSS As String
    Dim regionFromSS As String
    country = NO_VALUE
    region = NO_VALUE
```

Example: **Nesting** (3)

```
Do While ((country <> "Canada") And _
          (country <> "USA") And _
          (country <> "Mexico"))
    country = InputBox("North American country: ")
    Do While ((region <> "British Columbia") And _
              (region <> "Alberta") And _
              (region <> "Saskatchewan") And _
              (region <> "Manitoba") And _
              (region <> "Ontario") And _
              (region <> "Quebec") And _
              (region <> "New burnswick") And _
              (region <> "Nova Scotia") And _
              (region <> "Prince Edward Island") And _
              (region <> "Newfoundland and Labrador"))
          region = InputBox("Province to count: ")
    Loop
Loop
```


Example: Nesting (4)

```
countryCount = 0
regionCount = 0
row = START_ROW
countryFromSS = Cells(row, COUNTRY_COLUMN)
Do While (countryFromSS <> NO_VALUE)
    regionFromSS = Cells(row, REGION_COLUMN)
    If (countryFromSS = country) Then
        countryCount = countryCount + 1
    End If
    If (regionFromSS = region) Then
        regionCount = regionCount + 1
    End If
    row = row + 1
    countryFromSS = Cells(row, COUNTRY_COLUMN)
Loop
```

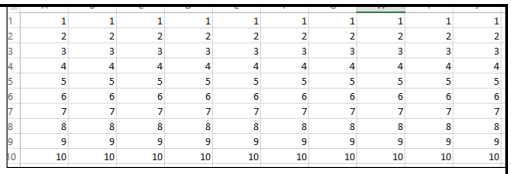
Example: Nesting (4)

```
        Range("F2") = "# clients from " & country
        Range("G2") = countryCount
        Range("F3") = "# in " & country & " who live in " & region
        Range("G3") = regionCount

    End Sub
```


(WB Ex 5) An Excellent Exercise To Help You Prepare For The Assignment: The Last Workbook Exercise

- The last exercise is quite challenging (you already have 3 graded components assigned which included the basics of VBA programming).
- Similar to the full assignment the exercise requires that you implement a solution using nested loops.
  - Workbook Exercise:
    - Outer loop to traverse from the start of the days where visitors came to town until the end.
    - Inner (nested) loop runs from start to finish each time the outer loop runs: traverses all the visitor information for a particular month.
  - Assignment:
    - Outer loop to traverse from the start of the Covid cases until the end (empty row).
    - Inner (nested) loop runs from start to finish each time the outer loop runs: traverses or steps through all the Covid cases for a particular day.

Exercise 4



1	1	1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7	7	7
8	8	8	8	8	8	8	8	8	8	8
9	9	9	9	9	9	9	9	9	9	9
10	10	10	10	10	10	10	10	10	10	10

- **Program description:**
  - **Using nested loops** the program will write the following information into the spreadsheet.
  - JT's comment: this one is substantially more challenging than the previous exercises but solving it will help you find a solution to the graded components.
  - Along row 1 from column 1 – 10 write the number 1 into sheet.
  - Along row 2 from column 1 – 10 write the number 2 into sheet.
  - Along row 3 from column 1 – 10 write the number 3 into sheet.
  - Continue along this pattern up to and including row 10 where the number 10 will be written.

Spreadsheet containing the solution (don't look at it until you have at least made an attempt): `Exercise4_nested_loops_numbering_cells`

VBA tutorial notes by James Tam

Exercise 5

- **Program description: get and display the month and year**
  - **Branch nested within a loop**
  - Prompt the user for a month as a numerical value from 1 – 12.
  - As long as value outside this range is entered the program will repeat the prompt.
  - After a valid value for the month has been entered the program will prompt for the day (again an integer value).
  - The program will repeatedly prompt for the day as long as a value outside the valid range has been entered.
    - The valid range depends upon the month:
      - February: ignore leap year and assume the maximum number of days is 28.
      - Month with 30 days: April, June, September, November
      - Months with 31 days: all other months

Spreadsheet containing the solution (don't look at it until you have at least made an attempt):`Exercise5_nested_loops_entering_month_day`

VBA tutorial notes by James Tam