CMP

# COMMERCE MENTORSHIP PROGRAM

## MIDTERM REVIEW SESSION

# COMM 205

Prepared by: Benjamin Gerochi

# TABLE OF CONTENTS



# Excel Basics

## Functions

- Preset formulas in Excel that can transform inputs (called **arguments**) to an **output**
- Example: =IF(B4 > 5, “Yes”, “No”)
  - IF is your function
  - B4 > 5, “Yes”, and “No” are your arguments
  - The output would be the result after hitting return

## Data Types

- Text: any **word(s)** or combination of characters and symbols
  - These need “quotation marks” if used as arguments
- Double: any **numbers** or numeric values
- Logical: any **TRUE or FALSE** value
  - Excel also sees 0 as False and 1 as True

## Operators

- Logical Operators
  - **=** (equal to)
  - **<>** (not equal to)
  - **>** (greater than)
  - **>=** (greater than or equal to)
  - **<** (less than)
  - **<=** (less than or equal to)
- Numeric Operators
  - **+** (plus)
  - **-** (minus)
  - * (times)
  - **/** (divided by)

# IF, NESTED IF

## IF

- Returns one of two possible values depending on whether a given logical argument is true or false

- Syntax:
  - **=IF(logical_test, [value_if_true], [value_if_false])**

- Arguments:
  - **logical_test:** the logical argument that is tested
  - **[value_if_true]:** the output if logical_test is true
  - **[value_if_false]:** the output if your logical_test is false

- Example:

| | A | B | C | D |
|---|---|---|---|---|
| 1 | **Name** | **Wins** | **Rank** | |
| 2 | Michael Gotbackup | 23 | =IF(B2>15, "Pro", "Noob") | |
| 3 | Lion Woods | 8 | | |
| 4 | Tyson Yes | 12 | | |
| 5 | The Pebble | 30 | | |

| | A | B | C |
|---|---|---|---|
| 1 | **Name** | **Wins** | **Rank** |
| 2 | Michael Gotbackup | 23 | Pro |
| 3 | Lion Woods | 8 | Noob |
| 4 | Tyson Yes | 12 | Noob |
| 5 | The Pebble | 30 | Pro |

# IF, NESTED IF

## NESTED IF

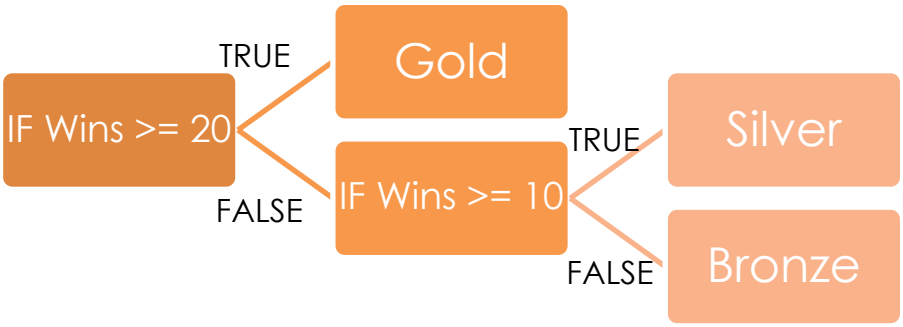
- An IF function within an IF function that is placed as either a value_if_true or value_if_false (2nd or 3rd argument)
- Used if we have **more than 2 possible values**

- Example:

| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| 8 | **Name** | **Wins** | **Rank** | | | |
| 9 | Michael Gotbackup | 23 | =IF(B9>=20,"Gold",IF(B9>=10,"Silver","Bronze")) | | | |
| 10 | Lion Woods | 8 | | | | |
| 11 | Tyson Yes | 12 | | | | |
| 12 | The Pebble | 30 | | | | |

| | A | B | C |
|---|---|---|---|
| 8 | **Name** | **Wins** | **Rank** |
| 9 | Michael Gotbackup | 23 | Gold |
| 10 | Lion Woods | 8 | Bronze |
| 11 | Tyson Yes | 12 | Silver |
| 12 | The Pebble | 30 | Gold |

- Drawing out a Tree Diagram can help you visualize the logic of what is occurring!




# AND, OR

## AND

- Tests whether **ALL** logical arguments are true
- Syntax: **=AND(logical1, [logical2], …)**
- Arguments:
  - **logical1, 2, …:** logical arguments being tested

- Example:

| | A | B |
|---|---|---|
| 1 | 20 | 10 |
| 2 | 40 | 30 |
| 3 | =AND(A1>B1, A2>B2) | |

| | A | B |
|---|---|---|
| 1 | 20 | 10 |
| 2 | 40 | 30 |
| 3 | TRUE | |

| | A | B |
|---|---|---|
| 1 | 20 | 10 |
| 2 | 30 | 40 |
| 3 | FALSE | |

- Became FALSE because not all of the arguments are now satisfied

## OR

- Tests whether **AT LEAST ONE** of logical arguments are true
- Syntax: **=OR(logical1, [logical2], …)**
- Arguments:
  - **logical1, 2, …:** logical arguments being tested

- Example:

| | A | B |
|---|---|---|
| 5 | 10 | 20 |
| 6 | 30 | 40 |
| 7 | =OR(A5>B5, A6>B6) | |

| | A | B |
|---|---|---|
| 5 | 10 | 20 |
| 6 | 30 | 40 |
| 7 | FALSE | |

| | A | B |
|---|---|---|
| 5 | 20 | 10 |
| 6 | 30 | 40 |
| 7 | TRUE | |

- Became TRUE because one argument is now satisfied

- Usually used in IFs or Nested IFs as the logical_test

# COUNTIFS, SUMIFS

## COUNTIFS

- Counts the number of values in a specified range/ranges that meet certain criteria/criterion

- Syntax: **=COUNTIFS(criteria_range1, criteria1, criteria_range2, …)**

- Arguments:
  - **criteria_range1, 2, …:** the range(s) which you want to test the criteria on
    - criteria ranges have to be of **equal length**
  - **criteria1, 2, …:** the certain criteria(s) that you want to be testing
    - this argument has to be enclosed in **"quotation marks"**

- Example:

| | A | B | C |
|---|---|---|---|
| 1 | Name | Wins | Tours |
| 2 | Michael Gotbackup | 23 | 6 |
| 3 | Lion Woods | 8 | 2 |
| 4 | Tyson Yes | 12 | 5 |
| 5 | The Pebble | 30 | 3 |
| 6 | =COUNTIFS(B2:B5, ">=10", C2:C5, "<=5") | | |

| | A | B | C |
|---|---|---|---|
| 1 | Name | Wins | Tours |
| 2 | Michael Gotbackup | 23 | 6 |
| 3 | Lion Woods | 8 | 2 |
| 4 | Tyson Yes | 12 | 5 |
| 5 | The Pebble | 30 | 3 |
| 6 | **2** | | |
| 7 | # of people that have at least 10 wins **AND** | | |
| 8 | have been on at most 5 tours | | |

- Note that it counts how many rows meet ALL criteria
- Counting the criteria separately: use =COUNTIFS() + COUNTIFS()
  - e.g. counting how many people have at least 10 wins **OR** have been on at most 5 tours

# COUNTIFS, SUMIFS

## SUMIFS

- Sums the values in a sum_range that meet certain criteria/criterion in criteria_range(s)
- Works similarly to COUNTIFS, except you are **adding** values that meet specified criterion
  - **ALL** criteria has to be met for a value to be added
  - criteria_ranges and sum_range have to be of **equal length**
  - criteria has to be enclosed in **"quotation marks"**

- Syntax: **=SUMIFS(sum_range, criteria_range1, criteria1, …)**

- Arguments:
  - **sum_range:** the range which you are adding up
  - **criteria_range1, 2, …:** the range(s) to test criteria on
  - **criteria1, 2, …:** the criteria you are testing

- Example:

| | A | B | C |
|---|---|---|---|
| 10 | **Name** | **Wins** | **Tours** |
| 11 | Michael Gotbackup | 23 | 6 |
| 12 | Lion Woods | 8 | 2 |
| 13 | Tyson Yes | 12 | 5 |
| 14 | The Pebble | 30 | 3 |
| 15 | =SUMIFS(B11:B14, C11:C14, ">=5") | | |

| | A | B | C |
|---|---|---|---|
| 10 | **Name** | **Wins** | **Tours** |
| 11 | Michael Gotbackup | 23 | 6 |
| 12 | Lion Woods | 8 | 2 |
| 13 | Tyson Yes | 12 | 5 |
| 14 | The Pebble | 30 | 3 |
| 15 | **35** | | |
| 16 | total wins of those who have been on at | | |
| 17 | least 5 tours | | |

# VLOOKUP

## VLOOKUP

- Looks up a value in the first column of the lookup table, and returns a value in the same row (based on a specified column)

- Syntax: **=VLOOKUP(lookup_value, table_array, col_index_num, [range_lookup])**

- Arguments:
  - **lookup_value:** the value you'll be searching for in the lookup table
  - **table_array:** the lookup table
    - usually locked using absolute referencing (see below) so table stays the same even when dragging down
  - **col_index_num:** the column which has your desired result
  - **[range_lookup]:** TRUE – approximate search, FALSE – exact

- Example:

| | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| 1 | **Name** | **Wins** | **Rank** | | | |
| 2 | Michael Gotbackup | 23 | =VLOOKUP(B2, $E$4:$F$9, 2, TRUE) | | | |
| 3 | Lion Woods | 8 | | | **Rank System** | |
| 4 | Tyson Yes | 12 | | | **0** | Noob |
| 5 | The Pebble | 30 | | | **5** | Bronze |
| 6 | Elon Doesnthaveto | 2 | | | **10** | Silver |
| | | | | | **15** | Gold |
| | | | | | **20** | Platinum |
| | | | | | **30** | Diamond |

| | A | B | C |
|---|---|---|---|
| 1 | **Name** | **Wins** | **Rank** |
| 2 | Michael Gotbackup | 23 | Platinum |
| 3 | Lion Woods | 8 | Bronze |
| 4 | Tyson Yes | 12 | Silver |
| 5 | The Pebble | 30 | Diamond |
| 6 | Elon Doesnthaveto | 2 | Noob |

# INDEX, MATCH

## INDEX

- Returns the contents of a cell in a specified row and column
- Syntax: **=INDEX(array, row_num, [col_num])**
- Arguments:
  - **array:** the range/table which contains your desired output
  - **row_num:** the row of your desired output (can use **MATCH**)
  - **[col_num]:** the column of your desired output
- Example:

| | A | B |
|---|---|---|
| 1 | **Name** | **Wins** |
| 2 | Michael Gotbackup | 23 |
| 3 | Lion Woods | 8 |
| 4 | Tyson Yes | 12 |
| 5 | The Pebble | 30 |
| 6 | =INDEX(A2:B5, 4, 1) | |

| | A | B |
|---|---|---|
| 1 | **Name** | **Wins** |
| 2 | Michael Gotbackup | 23 |
| 3 | Lion Woods | 8 |
| 4 | Tyson Yes | 12 |
| 5 | The Pebble | 30 |
| 6 | **The Pebble** | |

## MATCH

- Returns the row # of a specified value (column # if horizontal array)
- Syntax: **=MATCH(lookup_value, lookup_array, [match_type])**
- Arguments:
  - **lookup_value:** the value whose row/column # you are seeking
  - **lookup_array:** the range which contains the lookup_value
    - note that this has to have a width of only 1 cell
  - **[match_type]:** 1 – less than, 0 – exact match, -1 – greater than
- Example:

| | A | B |
|---|---|---|
| 8 | **Name** | **Wins** |
| 9 | Michael Gotbackup | 23 |
| 10 | Lion Woods | 8 |
| 11 | Tyson Yes | 12 |
| 12 | The Pebble | 30 |
| 13 | =MATCH("The Pebble",A9:A12, 0) | |

| | A | B |
|---|---|---|
| 8 | **Name** | **Wins** |
| 9 | Michael Gotbackup | 23 |
| 10 | Lion Woods | 8 |
| 11 | Tyson Yes | 12 |
| 12 | The Pebble | 30 |
| 13 | **4** | |

# LEFT, RIGHT, MID, CONCATENATE

## LEFT

- Returns the left of a cell based on a specified number of characters
- Syntax: **=LEFT(text, [num_chars])**
- Arguments:
  - **text:** the cell whose left portion you want to extract
  - **[num_chars]:** the number of characters you want to extract
- Example:

| | A | B |
|---|---|---|
| 1 | **Name** | **Prefix** |
| 2 | Mr. Michael Gotbackup | =LEFT(A2, 3) |
| 3 | Ms. Lion Woods | |
| 4 | Dr. Tyson Yes | |
| 5 | Mr. The Pebble | |

| | A | B |
|---|---|---|
| 1 | **Name** | **Prefix** |
| 2 | Mr. Michael Gotbackup | Mr. |
| 3 | Ms. Lion Woods | Ms. |
| 4 | Dr. Tyson Yes | Dr. |
| 5 | Mr. The Pebble | Mr. |

## RIGHT

- Returns the right of a cell based on a specified number of characters
- Syntax: **=RIGHT(text, [num_chars])**
- Arguments:
  - **text:** the cell whose right portion you want to extract
  - **[num_chars]:** the number of characters you want to extract
- Example:

| | A | B |
|---|---|---|
| 7 | **Name** | **Suffix** |
| 8 | Michael Gotbackup Jr | =RIGHT(A8, 2) |
| 9 | Lion Woods IV | |
| 10 | Tyson Yes Sr | |
| 11 | The Pebble VI | |

| | A | B |
|---|---|---|
| 7 | **Name** | **Suffix** |
| 8 | Michael Gotbackup Jr | Jr |
| 9 | Lion Woods IV | IV |
| 10 | Tyson Yes Sr | Sr |
| 11 | The Pebble VI | VI |

# LEFT, RIGHT, MID, CONCATENATE

## MID

- Returns the middle portion of a cell, given a starting point and a number of characters to extract
- Syntax: **=MID(text, start_num, num_chars)**
- Arguments:
  - **text:** the cell whose middle portion you want to extract
  - **start_num:** the numeric position of the character where you want to start extracting (is inclusive of this character)
  - **[num_chars]:** the number of characters you want to extract
- Example:

| | A | B |
|---|---|---|
| 13 | **Name** | **First Name Initial** |
| 14 | Mr. Michael Gotbackup | =MID(A14, 5, 1) |
| 15 | Ms. Lion Woods | |
| 16 | Dr. Tyson Yes | |
| 17 | Mr. The Pebble | |

| | A | B |
|---|---|---|
| 13 | **Name** | **First Name Initial** |
| 14 | Mr. Michael Gotbackup | M |
| 15 | Ms. Lion Woods | L |
| 16 | Dr. Tyson Yes | T |
| 17 | Mr. The Pebble | T |

## CONCATENATE

- Merges together the contents of given cells
- Syntax: **=CONCATENATE(text1, [text2], …)**
- Arguments:
  - **text1, 2, …:** the cells you want to merge together
- Example:

| | A | B | C | D | C |
|---|---|---|---|---|---|
| 19 | **First Name** | **Last Name** | **Full Name** | | **Full Name** |
| 20 | Michael | Gotbackup | =CONCATENATE(A20, " ", B20) | | Michael Gotbackup |
| 21 | Lion | Woods | | | Lion Woods |
| 22 | Tyson | Yes | | | Tyson Yes |
| 23 | The | Pebble | | | The Pebble |

# LEN, TRIM, SUBSTITUTE, REPLACE

## LEN

- Returns the number of characters in a particular cell
- Syntax: **=LEN(text)**
- Arguments:
  - **text:** the cell whose number of characters you want to count
- Example:

| | A | B |
|---|---|---|
| 1 | **Name** | **Length** |
| 2 | Mr. Michael Gotbackup | =LEN(A2) |
| 3 | Ms. Lion Woods | |
| 4 | Dr. Tyson Yes | |
| 5 | Mr. The Pebble | |

| | A | B |
|---|---|---|
| 1 | **Name** | **Length** |
| 2 | Mr. Michael Gotbackup | 21 |
| 3 | Ms. Lion Woods | 14 |
| 4 | Dr. Tyson Yes | 13 |
| 5 | Mr. The Pebble | 14 |

## TRIM

- Removes all spaces except single spaces between words
- Syntax: **=TRIM(text)**
- Arguments:
  - **text:** the cell that you want to trim
- Example:

| | A | B |
|---|---|---|
| 7 | **Name** | **Trimmed Name** |
| 8 | Mr. Michael Gotbackup | =TRIM(A8) |
| 9 | Ms. Lion Woods | |
| 10 | Dr. Tyson Yes | |
| 11 | Mr. The Pebble | |

| | B |
|---|---|
| | **Trimmed Name** |
| | Mr. Michael Gotbackup |
| | Ms. Lion Woods |
| | Dr. Tyson Yes |
| | Mr. The Pebble |

# LEN, TRIM, SUBSTITUTE, REPLACE

## SUBSTITUTE

- Substitutes an old specified text with a new specified text

- Syntax: **=SUBSTITUTE(text, old_text, new_text, [instance_num])**

- Arguments:
  - **text:** the cell which contains the text you want to substitute
  - **old_text:** the old text which you want to substitute
  - **new_text:** the new text which you want to substitute to
  - **[instance_num]:** the instance # of the old text to be substituted
    - if not indicated, will substitute all instances of the old text

- Example:

| | A | B | C |
|---|---|---|---|
| 13 | **Name** | **Substituted Name** | |
| 14 | Michael Gotbackup Jr. | =SUBSTITUTE(A14, "Jr.", "The Goat") | |
| 15 | Lion Woods Jr. | | |
| 16 | Tyson Yes Jr. | | |
| 17 | The Pebble Jr. | | |

| | A | B |
|---|---|---|
| 13 | **Name** | **Substituted Name** |
| 14 | Michael Gotbackup Jr. | Michael Gotbackup The Goat |
| 15 | Lion Woods Jr. | Lion Woods The Goat |
| 16 | Tyson Yes Jr. | Tyson Yes The Goat |
| 17 | The Pebble Jr. | The Pebble The Goat |

# LEN, TRIM, SUBSTITUTE, REPLACE

## REPLACE

- Replaces content in a cell with new specified text, given a starting point and number of characters to replace

- Syntax: **=REPLACE(old_text, start_num, num_chars, new_text)**

- Arguments:
  - **old_text:** the cell that contains the content you want to replace
  - **start_num:** the numeric starting position of the content you want to replace
  - **num_chars:** the number of characters to be replaced
    - this is not necessarily equal in length to the new_text
  - **new_text:** the new text which you want to replace to

- Example:

| | A | B |
|---|---|---|
| 19 | **Name** | **Replaced Name** |
| 20 | Mr. Michael Gotbackup | =REPLACE(A20, 5, 2, "No") |
| 21 | Ms. Lion Woods | |
| 22 | Dr. Tyson Yes | |
| 23 | Mr. The Pebble | |

| | A | B |
|---|---|---|
| 19 | **Name** | **Replaced Name** |
| 20 | Mr. Michael Gotbackup | Mr. Nochael Gotbackup |
| 21 | Ms. Lion Woods | Ms. Noon Woods |
| 22 | Dr. Tyson Yes | Dr. Noson Yes |
| 23 | Mr. The Pebble | Mr. Noe Pebble |

# FIND, SEARCH

## FIND

- Finds the numeric position of a specified text, and can start finding at an optional point (e.g. if you don't want to find the first instance)

- Case sensitive
- Does not allow for wildcards (?, *, ~) (see SEARCH)

- Syntax: **=FIND(text, within_text, [start_num])**

- Arguments:
  - **text:** the text whose numeric position you want to find
  - **within_text:** the cell that contains the position you want to find
  - **[start_num]:** the numeric position where you want to start finding
    - e.g. if you want to find the second instance, your start_num could be FIND(first instance) + 1

- Example:

| | A | B | B |
|---|---|---|---|
| 1 | **Name** | **Find E** | **Find E** |
| 2 | John Mayer | =FIND("e", A2, 6) | 9 |
| 3 | Michael Jackson | | 6 |
| 4 | Aubrey Drake Graham | | 12 |
| 5 | Kanye West | | 8 |

# FIND, SEARCH

## SEARCH

- Searches for the numeric position of a specified text, and can start searching at an optional point. Similar to FIND, except:

- Is not case sensitive
- Can allow for **wildcards** (?, *, ~)
  - can substitute for unknown characters in text
  - **?** – substitutes **any 1** character
    - e.g. b?o can search "bro" or "boo"
  - ***** – substitutes **any number** of characters
    - e.g. t*s can search "toes", "toss", "ts", or "two shoes"
  - **~** – placed before a wildcard to **treat it as actual text**
    - e.g. you want to find the text "?" specifically
    - SEARCH("?", …) will treat ? as a wildcard/search anything
    - SEARCH("~?", …) will find "?" specifically

- Syntax: **=SEARCH(text, within_text, [start_num])**
  - see FIND for argument definitions

- Example:

| | A | B | C | B |
|---|---|---|---|---|
| 7 | **Name** | **Search E** | | **Search E** |
| 8 | John Mayer | =SEARCH("E", A8, 6) | | 9 |
| 9 | Michael Jackson | | | 6 |
| 10 | Aubrey Drake Graham | | | 12 |
| 11 | Kanye West | | | 8 |