

# Objectives for each module are

---

## 02: Computers 101 + Developing A Solution (1) & 03: Developing A Solution (2)

- Understand the basics concepts of a computer: the hardware, the software, the different languages.
- Vocabulary:
  - Software
  - Hardware
  - Programming language
  - Von Neumann architecture
  - CPU
  - Operating System
  - RAM
  - Volatile

## 04: Starting MATLAB & Variables

- Get finally acquainted with MATLAB.
- Where/what is the command window?
- Where/what is the workspace?
- Where/what is the current folder?
- Where/what is the history window?
  
- What are script files?
- How to create a script file?
  
- What are variables?
- How are variables useful?
- Creating variables?
- Assigning values to variables?

## 05: Vectors & Plotting

- What is a vector in physics?
- Give at least two examples of physics quantities that are vectors.
- Be able to write in vector form a vector given graphically.
- Be able to code a vector in MATLAB.
- Recognize the mathematics behind vector addition.
- Be able to add vectors graphically.
- Can you multiple two vectors together?
- Can you divide two vectors together?
- Recognize when element-per-element operators are required

## 06: Data Types & Input and Output

- VARIABLES & DATA TYPES:
  - Understand that variables are behind everything in coding.
  - Understand how values are stored in memory.
- INPUTS & OUTPUTS:
  - Give the user the ability to enter numbers each time the code runs.
  - Give the user the ability to enter words.
  - Show results to the screen.
    - for the purpose of debugging (i.e. finding errors in the code)
    - to show values only
    - to format a sentence that shows results

## 07: Operators & Conditional (if)

- CONDITIONALS:
  - Understand 0 and 1. False and True.
  - Recognize how to write a condition so MATLAB can decide properly what to do.
  - Know the syntax of all operators.
    - Relational Operators
    - Boolean/Logical Operators
    - Arithmetic Operators
- if STATEMENTS:
  - Understand that not ALL lines must always be executed.
  - Know the syntax of an if statement:
  - What keywords are mandatory
  - What keywords are optional
  - What order do the keywords go in?

## 08: Conditionals (switch)

- Know the keywords of the switch statement, and their order.
- Recognize which keywords are necessary.
- Recognize when a switch statement is easier to code than an if statement
- When possible, easily rewrite an if into a switch , and vice-versa.

## 09: Library Functions, Random, Rounding

- Recognize the vocabulary words.
- Understand how a function works overall.
- Immediately figure out the equation to generate a random value between 2 specific limits.
- Recognize the keywords that can round up/down/w.r.t 0.5

## 10: Intro to Loops

- Recognize when a loop is useful.
- Know the two vocabulary words for both loops.
- Identify the criteria that make a loop work properly.
- Write a while loop to trap the use.
- Indent properly a code according to conventions.
- Write a running total.
- Interrupt an infinite loop.
- Read a loop and indicate what it does
- Develop an algorithm to solve a problem using a while loop.

## 11: Loops (User Input Vectors)

- Know by heart when to choose for vs. while.
- Don't hesitate between for and while.
- Write each syntax of a for loop and a while loop.
- Switch easily a while loop into a for loop.
- Develop an algorithm to solve a problem using a for loop.
- Understand running totals (whether it is running sum, or running product)
- Set up a running total (start of the total, sum/product)
- Set up a for loop to populate a vector

## 12: Loops (Traversing Vectors)

- Recognize some real-world applications of loops.
- Learn at least 2 mathematical methods that are considered numerical methods.

## 13: Intro to Arrays (1)

- Recognize when an array is useful.

## 14: Arrays (2)

- Creating scalars, row vectors, column vectors, transposing, matrices
- Using keywords on arrays
- Understand how operations on arrays happen
- What is "array augmentation"?
- Referencing elements of a vector
- Referencing elements of a matrix
- Operations on vectors and matrices give different results.
- Slicing arrays
- Concatenating (to link together) values into vectors, into matrices

## 15: High-Level Files & Logical Operations

- Be capable of seeing how software uses a file.
- Realize where files are needed for daily software, such as agenda on a phone.
- Recognize files that can be read using `dlmread()` vs. other functions.
- Understand logical operations, and the use of them with arrays.
- Prefer using logical operations instead of writing loops.

## 16: Excel & Cell-Arrays

- Know the syntax of `xlsread()`. Not necessarily by heart, but quickly recall it from the help file.
- Understand how `xlsread()` works, and how it separates the data.
- Choose the appropriate dialog box for your software.

## 17: Functions (1)

- Recognize how functions are used in today's world.
- Be able to decide which file is a:
  - function definition
- and which file has the:
  - function-call
- In between those two above, be able to pick out:
  - the parameters, (in function definition)
  - the return-values, (in function definition)
  - the arguments, (in function call)
  - collecting values, (in function call)
- Know by heart the layout of writing a function.
- Understand & explain the unique way that data flows in and out of a function.

## 18: Functions (2)

- Continue learning the layout of writing a function.
- Get comfortable writing functions that use multiple inputs and return multiple values.
- Remember that there is an order to the variables in the call when a function returns more than 1 value.
- Understand how to ignore return values when the program does not need them.

## 19: GUIDE

- Purely if you want to be introduced to Graphical User Interface. OPTIONAL. (Veterans' day)

## 20: Strings

- Know how to hardcoding strings
- Know how to prompt the user for strings
- What is a string?
- Recognize and use the few built-in functions used in strings
- Know how to compare strings together (and how NOT to!)
- Concatenate strings horizontally and vertically
- Slice Strings
- Create dynamic prompts

## 21: Low-Level Files

- Recognize whether a file can use high-level functions or does it require low-level
- Know the order of the commands needed to open, read, and close a file

## 22: Final Project due. Introduction to a compiled language

- Recognize how compiled languages work.