**May 2019 Workshops**

**Saturday (4 sessions of 80 minutes each)**

1. **Hands-on databases: design a Relational Database using MS Access (80’)**
   1. Importing data from Excel spreadsheets
   2. Working with Tables
   3. Understanding Data Dictionaries and setting them up.
   4. Creating relationships: One-to-One and One-to-Many.
   5. Creating Forms
   6. Creating Queries and from the Queries Reports
   7. Creating a Graphical user Interface to manage the database.
   8. Example of Assessment Task: downloading raw data, applying data verification criteria, reshaping data in two Excel spreadsheets and then following “a” to “g” steps.
2. **Hands-on Animations using Animate CC (80’)**
   1. Use of key frames and onionskin.
   2. Cell-by-Cell animation.
   3. Motion and Shape tween (morphing).
   4. Path guided motion tween.
   5. Hand out of resources and exercises to use all of the above features.

**Resources and class assessment examples will be handout at the end of this session.**

1. **Hands on IPO charts, Pseudocodes, Flowcharts and Deskchecking (80’)**
   1. Using all coding structures, including: sequence structure, pre-test repetition, post-test repetition, multiway selection, counted loops and binary decisions.
   2. Dataset selection and deskchecking.
   3. Applying all of the above in different cases scenarios.

**Resources will be handout at the end of this session.**

1. **Hands on Coding Using Scratch (80’)**
   1. Introduction to Scratch: creating a simple “ice breaker” program (“pinball”)
   2. Creating a basic program to select different shapes and draw them on the screen.
   3. Using Scratch to resolve simple equations: linear equations and quadratic equations.
   4. Create a simple game with increasing level of difficulty to the user and with a score.

**Completed programs will be handout at the end of this session**

**Sunday (3 sessions of 80 minutes each)**

1. **Python Coding Fundamentals – Part I (80’)**
   1. Use of the IDE (Integrated Development Environment), Data Types
   2. Sequential structures, binary decision structures using IF THEN ELSE loops, Counted loops using FOR loops, pre=test loops using WHILE loops
   3. Translating Pseudocodes into Python Codes. Coding Examples: quadratic equations, temperature conversion, applying Python to solve basic problems..
2. **Python Code – Special Features – Part II (80’)**
   1. Introduction to OOP (Object Oriented Programming: use of functions, advantages of modular programming.
   2. Coding and problem solving examples using modular programming approach: build a calculator, convert the temperature conversion done in Section I using a modular approach. Fibonacci series problem solved using a modular approach.
   3. Introduction of exceptions: capture of exceptions such as wrong data type and division by zero.
   4. Final project (if time allows): code a solution to an electrical circuit where the user can choose between Series of Parallel circuit, 2 or 3 resistors, and using the total current and individual resistors values, it calculates the total resistance of the circuit and the total voltage.