Report for the Unit EE30241.

At submission (including the code) a report containing the description, explanation and reflection on the different exercises is required (max 3 pages per exercise, including pictures and references).

Each exercise "chapter" will have a structure as follows:

- **Introduction including a description of the task**
- **Solution explanation: which include a description of the code (only a few lines can be used when relevant for discussion) screen shots, data if any – hint: for example, in ex1 description of the protocol (commands, parser etc…)**
- **Conclusion: which includes reflection on the specific exercise and what could have been improved, what went wrong and what was good.**

Keep a logbook to help in writing the report at the end of the unit.

A Demo Video will be uploaded during the submission. Further details on the submission Instructions document.

Exercise 2 – EE30241

This exercise is part of the marked coursework. The Report part of the Coursework will contain the reflection and description as described above.
Exploiting what you learnt from Ex1 build an enhanced ROS version of such calculator system. Create a ROS package/system with **at least 3 nodes**. The calculator will have the functionalities as in Ex 1, but in this case, each node will have a specific task (use messages, services and other ROS features as needed).

(#1) this node will be the user input/output interface (text based) sending commands to the calculation node and log node (#2) and a third node (#3) which will provide timestamp of running time to (#1). Calculation instruction are passed in a similar way as in Ex 1. The commands as well as if the expression is wrong or there is an error those need to be logged using ROS logging functionalities. Exceptions needs to be handled.

(#2) this node will handle the calculations and share info with nodes #1. This node will also notify node #1 about how many operations have been performed counting them. This node will also log the transactions and errors. Node #1 will be able to clear this log with a command or, with another, recall a list of the history and with another will be able to recall a specific line to run the calculation again. At start up the log will remain and consequently node #1 will need to be notified of, for example, the number of elements in the log.

(#3) this node is a timekeeper. It advertises a message with time stamp and running time of the program to #1 (to be displayed on GUI or command will print the latest value) and #2 (to log current time stamp for each entry). The log history of commands and results will be saved on a file.

Provide a launch file to execute the program and a README.txt on how to use the package and definition of main commands to use/functionalities e.g. if need to run only the launch file or if other actions are needed.

Ideal steps Ex2:

A) Use you own shared drive where possible.
B) Create ROS workspace (it should be already set up on your machines – if not call the folder ros_ws_loc).
C) Create ROS package (be sure folder src within ros_ws_loc is present or create one)
D) Create ROS nodes
E) Use one node to handle the operations (min: = + - * / ^ sqrt) and logging.
F) Use one node to be the user interface (GUI or text based)
G) Use one node to keep time and generate the time stamp messages.
H) Write all the needed nodes / messages / services / actions / launch files