THIS IS A PROCTORED PRACTICAL

YOU MUST SHARE YOUR SCREEN SO YOUR PARTICIPATION IN THIS PRACTICAL CAN FULLY INVIGILATED

1. Create a Github repository "Assembly\_and\_C"
2. Create a sub directory PRACTICAL\_##
3. Add Github link to CA Spreadsheet   
   e.g <https://STUDENTID.github.com/Assembly_and_c/PRACTICAL_##>
4. Invite Lab Supervisors including **MuddyGames** as a collaborators
5. Go to designated group to complete practical
6. Upload completed Practical files to Github repository

NOTE: Use of EASy68K editor and emulator allowed, use of internet allowed, use of slide deck(s) allowed. Installer located here <http://www.easy68k.com/>

Create a unique file ***e.g. practical\_##\_part#.X68*** for each practical section below.

**Objective** Understand and utilise Basic Memory concepts, BINARY, HEX and Literals**:**

|  |  |  |
| --- | --- | --- |
| **1** Create a new 68K project and name the file ***practical\_02\_part1.X68***  Edit compile and execute the code below, examine and note contents of data registers and memory. | | |
|  | | |
| **2** | Create a new 68K project and name the file ***practical\_02\_part2.X68***  Edit compile and execute the code across and observe contents of memory while debugging.  Review questions, what do the numbers stored in memory mean and why? |  |
| **3** | Create a new 68K project and name the file ***practical\_02\_part3.X68***  Edit compile and execute the code across and observe contents of memory while debugging.  Review questions, what do the numbers stored in memory mean and why? |  |
| **4**  **Bytes can store 8 bits**  **Word can store 16 bits**  **Longword stores 32 bits**  **You can overwrite parts of word of long with bytes** | Create a new 68K project and name the file ***practical\_02\_part4.X68***  Edit compile and execute the code across and observe contents of memory while debugging (based on the **Specification**)  Review questions, what do the numbers stored in memory mean and why?  Examine what are the values that can be stored for each bit width of data? | **4 Specification**: Complete the following operations  MOVE.L  MOVE.B  MOVE.W  Moving data to memory  Use your own examples  For Literals, Hex and Binary  Move to **Data** to **Data Registers** and to **Memory**. Move data from **Memory** to **Data Registers**.  Your solution should include at least **18 examples** for the above. |
| **5** | Complete Practical Quiz which will be provided by Lab Supervisor | |

**Demonstrate completed assembly files at the end of the LAB and ensure it has been checked**

|  |  |  |  |
| --- | --- | --- | --- |
| **Student Name** | **Karolis Grigaliunas** | **Student Number** | **C00287940** |
| **Date** | **22/01** | **Checked** |  |