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 Arithmetic, Logic Operations and BSR and BRA Branching**:**

|  |  |  |
| --- | --- | --- |
| **1** | Create a new 68K project and name the file ***practical\_04\_part1.X68***  Edit compile and execute the code across and observe while debugging and contents of memory.  Examine and note contents of address registers and memory.  Review questions, what do the address register mean and what is stored in memory and why?  The AR (A1) holds $3000 which is the players hp. It takes 20 hp when hit by npc and adds 10 when hp pack found |  |
|  |
|  |
| **2** | Create a new 68K project and name the file ***practical\_04\_part2.X68***  Edit compile and execute the code across and observe while debugging and contents of memory.  Examine and note contents of address registers and memory.  Write examples for OR, NOT and EOR (Exclusive OR)  Review questions, what do the logic instructions mean, what Addressing Modes are used?  The addressing modes used are direct value addressing and register addressing.  The logic instructions mean as they would in digital Logics.  AND if both operands are 1 the result will be 1.  OR one of the bits must be 1 or else the result will be 0  NOT inverts the bits, 0 becomes 1 and 1 becomes 0  XOR the result will be 1 if the bits of the operands are different |  |
| **3** | Create a new 68K project and name the file ***practical\_04\_part3.X68***  Edit compile and execute the code across and observe while debugging and contents of memory.  Examine and note contents of address registers and memory.  Review questions, what are the Branch Instructions useful for BRA and BSR, what Addressing Modes are used? |  |
|  |
|  |
| **4** | Create a new 68K project and designate the file as ***practical\_04\_part4.X68***.    Perform tasks such as editing, compiling, and executing the code according to the **Specification**. During debugging, closely monitor the contents of the memory.  Inspect the stored values in memory and adjust values and their locations within the memory.  Review questions what Addressing Modes are used?  Direct Addressing | **4 Specification**: Complete a simple game that uses   * Data Registers * Address Registers * Arithmetic Operations (ADD and SUB) * Logical Operation (AND, OR and EOR) * Branch Instructions (BRA and BSR)   Declaring initial data such as Player Health at 100% or $64 (Hex equivalent)  Use your own examples, such as typical game data;   * Player Points * Player Health * Player X and Y Position * Boss Health * Boss X and Y Position   Modify the Data Values during programme operation |
| **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** | **Ariel Fajimiyo** | **Student Number** | **C00300811** |
| **Date** | **05/02/25** | **Checked** |  |