**Part A – Documentation**

**Due Date:** (16/09/2022)

* **Part A:** consists of your project documentation and a test prototype.

This document will include an outline of your program structure as a **flowchart and UML Class diagrams.**

The assignment documentation must be created as a **single Word or PDF document**.

**Your document should contain the project flowchart and the clearly identified UML diagrams of all your proposed Classes. You can use the *Insert > Table* option in Word to create the UML diagrams or you can use any other software, provided that the diagrams are included as *legible images* in your submitted document.**

Your project prototype must be submitted as a Visual Studio (Code) project, including all header and definition files, and any appropriate text files to ensure the program compiles and runs.

Explicit assessment criteria are provided, however please note you will also be assessed on the following broad criteria:

* + Details of a proposed project plan for the overall project.
  + Creating accurate and complete UML diagrams.
  + Applying a solid Object-Oriented Design (OOD) for the overall project.
  + Using appropriate naming conventions.

Text

Description automatically generated

**Your flowchart should include:**

* The game setup (everything that happens before the game starts
* The player’s turn (the sequence of events that happen during a turn)
* Processing player input (the commands your player can use)
* Providing feedback to the player (in response to the player’s interactions)
* The end game conditions (include all win and lose conditions)
* Include the functionality of all your game classes – see Part B
* Additional Features included, if any – see Part B

Here is an example to get you started with and algorithm that you can convert into your flowchart:

1. The Game Setup
   * Read game rules from text file and display
   * Add the player – ask for the player’s name, set default variables
   * include all the other things that will happen during initialisation including
     + creating the hazards and the game world
     + initialising other game variables

As you can see, you only have to describe the actions the program will take, not the code. The idea here is to give you a starting point for when you start writing your code as you can use this as a checklist of the things you need to include.

**READ THE REST OF REQUIREMENTS IN “FULL C++ Project File” (PROJECT TEST PLAN, UML DIAGRAMS, MAP YOU WOULD INCLUDE, THE PROTOTYPE, ETC)**

**Project Documentation [50] \*\*\*\*\*USE THIS AS A CHECKLIST FOR THE DOCUMENTATION ALONE\*\*\*\*\***

* Flowchart Design **[15]**
  + Has used the correct flowchart symbols [1]
  + Has used appropriate descriptors in flowchart symbols [1]
  + Flowchart includes all game functionality (Parts A & B) [2]
  + Flowchart clearly shows the correct sequence of the game (setup, gameplay, end game) [4]
  + Pre-defined processes include all the required game functionality [5]
  + Processes are performed in a logical order [2]
* UML Class Diagrams **[10]**
  + Correct structure used (Name, Attributes, Behaviours) [1]
  + Included the correct designations for public [+] and private [-] data [1]
  + All variables and functions have meaningful names [1]
  + Included constructor(s), destructor, and appropriate access and mutator functions in all custom classes [4]
  + Included a class diagram for the Player, Monster, Location, and Item classes and the Main (application) [3]
* Project Test Plan **[20]**
  + Each class tests all constructors, accessor and mutator functions [12]
  + Appropriate testing procedures for selecting options, combat and end game conditions [8]
* Other Documentation **[5]**
  + The introduction to your game as a description (own words) [3]
  + Included a map with location names [2]

Complete tasks here: (of course, create as many pages as you need)