# Achievement Standard

| **Number** | **Version** | **Title** | **Credits** | **Assessment** |
| --- | --- | --- | --- | --- |
| AS91906 | 1 | Use complex programming techniques to develop a computer program | 6 | Internal |

# Assessment Criteria

| **Achievement** | **Achievement with Merit** | **Achievement with Excellence** |
| --- | --- | --- |
| Use complex programming techniques to develop a computer program. | Use complex programming techniques to develop an informed computer program. | Use complex programming techniques to develop a refined computer program. |

# Achievement Level (Construct)

To meet the Achieved Level Criteria you are required to write a computer program that meets the specifications of an agreed brief which is provided to you in this document.

In addition, you need to ensure that your solution adheres to the requirements of the standard, paying particular attention to incorporate all of the required components, some of which are listed below.

* Writing code for a program that performs a specified task.
* Using complex techniques in a suitable programming language.
* Set out the program code clearly and documenting the program with comments.
* Test and debug the program to ensure that it works on a sample of expected cases.

# Achievement with Merit Level (Develop Informed)

In addition to the achievement level requirements, you are also expected to demonstrate your ability to …

* Document the program with **appropriate variable/module names and organised comments that describe code function and behaviour.**
* **Follow common conventions** for the chosen programming language.
* Test and debug the program in an organised way to ensure that it works on a sample of **both expected cases and relevant boundary cases**.

# Achievement with Excellence Level (Develop Refined)

In addition to the merit level requirements, you are also expected to demonstrate your ability to …

* Ensure that the program is a well-structured, logical response to the specified task.
* Make the program flexible and robust.
* Comprehensively test and debug the program.

**How do I do this?**

Complete the Programming documentation sheets provided.

* Section 1: This will get you to evidence the coding requirements to meet the Achievement Level criteria.
* Section 2: This will get you to evidence the coding requirements to meet the Merit Level criteria.
* Section 3: This will get you to evidence the coding requirements to meet the Excellence Level criteria.

**AND**

* Section 4: Complete the relevant sections to demonstrate and test your program to the appropriate level.

# Examples of Complex programming Techniques include:

* Programming or writing code for a graphical user interface (GUI).
* Reading from, or writing to, files or other persistent storage.
* Object-oriented programming using class(es) and objects defined by the student.
* Using third party or non-core API, library or framework.

For more information on the technical expectations for this assignment please refer to the documentation for this standard.

# Iterative Process?

This standard does not require you to adopt an iterative development process, but recommends that you do.

It is perfectly possible to meet the Excellence level criteria with just a single submission. However, each of the documentation sheets provided gets you to focus on particular areas that will aid in your overall achievement, so it is highly recommended that you use these.

In the event that you opt to not submit multiple iterations of this project, please take the time to ensure that your submission meets **and documents** the expectations found in these three worksheets.

# Brief: Dungeon Delving Game 2

Remember the Dungeon delving game from 2022? Well, it’s back, but this time on a more developed scale.

You have been asked to create a digital dungeon delving game using a programming language of your choice. This will be a single player game where you choose to either allow your character to either delve deeper into the dungeon or go home. If you choose to delve deeper then you may engage an enemy and need to fight. You may be lucky and defeat it, potentially adding to your score, or you may be beaten and lose the game with no score. If you choose to go home the game ends but your score is safe.

# Basic Instructions (Be careful, these are a little different this time!)

1. Create a hero and award them a number of hit-points (HP).
2. Each time the player moves forward there is a chance they will encounter a random enemy.
3. Each enemy has a name and 3 values used to generate the following 3 statistics.
   * + Their Hit Points (generated once when the monster is encountered)
     + Gold (generated once when the monster is encountered)
     + Damage Dealt (generated each battle round)
4. When encountering an enemy, the player and enemy both deal and take damage until one of them is defeated or the player attempts to leave the battle.
5. If the enemy is defeated (HP <= 0), the player received the enemy’s gold.
6. If the player is defeated (HP <= 0) , they lose everything. Game over!
7. If the player attempts to leave the battle, then the enemy gets to attack them one more time.
8. The player also has the ability to use a ‘heal’ option a fixed number of times each game.
9. The amount of gold collected is the players score in the game.

# Advanced Options (examples, you can add more!)

To make the game more interesting the following additional features could be added. These are not required but may provide you with a greater opportunity to demonstrate your programming ability as well as making the game more interesting.

* Add the possibility of the hero encountering something else.

E.g.

* + Find an additional healing potion (up to a set limit) **\***
  + Find an invisibility potion (to avoid an enemy completely)
  + Find a weapon to create additional dealt damage. **\***
  + Find some armour to reduce the damage taken. **\***
* Provide a chance that equipment gets broken or damaged.

E.g.

* + Potion vial gets smashed.
  + Weapon or armour gets damages reducing effectiveness. **\***
* Make the consequences of fleeing a battle more significant. E.g. Chance of losing some gold.**\***
* Let the player choose a difficulty mode that provides them with different challenges. E.g. a greater chance of encountering an enemy, more challenging enemies etc.
* Offer the option to play again when the game is finished.
* The game could allow for an additional multi-player variation where a winner is declared as the player who collects the most gold.
* A Leaderboard could be included to display the highest achieving players.
* Media elements such as images and sound cold be included.

\* Included in example game

# Complex Skill Opportunities

* Enemies, Equipment, Players could all be managed by classes.
* Game Environment could use a GUI.
* Saving Progress, a Leader board and loading enemies details would each require persistent storage.

# Provided Resources / Examples

To assist you in getting started, the following materials have been provided to you to explore if you wish. There use is not a requirement of the assignment.

* **Example Completed Game** (.exe and video): This is an example solution and not a blueprint that you need to try and emulate if you do not want. This includes some extended functionality but is fairly limited in other ways.
* **Example Enemy data file**: Each enemy has 4 values.
  + Name: Naming or describing the enemy
  + Health: the number of 6 sided dice rolls used to create their health at the start of a battle.
  + Damage: the number of 6 sided dice rolls used to create the damage they inflict – recalculated each round.
  + Gold: the number of 6 sided dice rolls used to create the gold treasure found by the player when the enemy is defeated.

# Authenticity

To ensure authenticity of the work presented, part of the assessment process for this assignment will require you to explain sections of your code identified by your teacher. This will be done via an interview and will help to both demonstrate your understanding of the code as well as help justify your choices.

Please note that this process DOES NOT negate the need for effective commenting of your code, which should also be completed as if it were for a third party (someone else who does not know what you program is supposed to do).