# Assignment 2 - Part 2 - COMP2152

## Part 2 - Explain your Code (50%)

Write the answers in the doc provided and save and submit Part 2 as a PDF. You do not have to write or submit any new code for this section. I want you to understand how you could work on a piece of code that already exists (as is the case when working with Open Source code), and how to improve it. You can **type** in your answers *or* complete it **by** **hand** (handwriting MUST be legible) and then scan your submission.

1. How have we used classes for our project to reuse code?  
   By using inheritance, code does not need to be rewritten multiple times (Hero & Monster class being extended from the Character class).
2. Provide 1 line of code, as one of many examples, where code is shared between the monster class and the hero class?  
   self.\_health\_points = random.randint(5, 20)
3. What is the benefit of using complex getters and setters?   
   They allow us to add transformation, validation or additional logic when modifying/accessing attributes, which protects data integrity and prevents unwanted modifications.
4. If we didn’t use try-except blocks, what would be the problem?

The program can crash due to unexpected runtime errors.

1. How could we use the name of the **operating system or the version of python** in your game to prevent errors? Choose just 1 of the above.  
   We could check if the python version is met. There can be a warning at the beginning if the minimum version is not met.
2. What’s another piece of information we could save inside of the save.txt file? (Remember, we load this information every time we start a new game, so that we can keep track of all of the games you have played so far. )  
   We can save the player’s progress so the player’s stats and equipment will stay the same as their last session.
3. **New Feature:**
   1. Think of **1** **new feature** you can add to the game that could use list comprehension and nested conditional statements. For now just write 1 sentence that describes the feature:  
        
      Now add your new feature description here:  
      A dynamic loot system where the loot is randomized based off of the level/difficulty of the monster (Boss monster vs. normal monster).

**Examples:**

Below are the examples to show you that you can be very creative, and you should have fun with this exercise. You must use an idea that is NOT directly on the list below:  
  
**eg a)** Add another monster so that the hero can fight 2 monsters at once

**eg b)** Create a digital board game, that shows the hero moving around to different towns on a map  
  
**eg c)** Add a dog that runs in front of the hero and discovers features about the world

* 1. Give the new feature you created a short 2-3 -word a title:   
       
     Now write your Title here:  
     Dynamic Loot System

**Examples:**  
**eg a**) Multiple Monsters  
**eg b)** Roam Towns  
**eg c)** Dog Scout

* 1. Explain how you could implement the idea you chose. You must explain how you would use both of the control structures below. Draw a diagram, map, sketch for each (you can use any software for this, e.g. Draw.io). You don’t have to match the style of diagram I have here, just use a visual to describe your idea. Note, you must have loops and conditional statements diagrammed below as needed:  
     1. **Using a list comprehension loop**  
        List comprehension will be used to create loot from a existing list of potential items, filtering out items that meet a certain requirement.

|  |  |
| --- | --- |
| Loot Items | Rarity |
| Enchanted Sword | Legendary |
| Coins | Rare |
| Health Potion | Common |

* + 1. **Using nested conditional statements**  
       If the hero slays a boss monster, they will receive Rare or Legendary items. Normal monsters will drop Common items.  
         
       A diagram of a game

       AI-generated content may be incorrect.

**Example:**  
**eg b)** Roam Towns

* + 1. **Using a list comprehension loop**  
       Every time in the loop, move one square in 1 direction, (N, E, S, W). Have a variable that keeps track of the Hero’s location by saving values of the board. We can have 2 nested for loops and store the map as a 2D array.  
         
       Eg.   
       Hero location is currently at Row 3, Column D.   
       Town 2 location is at Row 4, Column G.   
       Town 1 location is at Row 1, Column A .  
         
       **Diagram:**  
         
        A B C D E F G H I  
       1 (town1\_loc)  
       2  
       3 (player\_loc)  
       4 (town2\_loc)
    2. **Using a nested Conditional Statement**  
       **If** the hero is in Town 2, **then** allow the hero to buy armor but not sell. Otherwise, the hero can sell armor but cannot buy.   
         
       Create an array of armor options available in Town 2. He could also trade some of his loot based on the value of the loot he has.

carbon fiber breastplate,   
steel helmet,   
 bionic gloves

armour\_town\_2

array

can\_sell

true/false

boolean

RAM

if hero is at town 1

hero can’t sell  
 hero can buy

Display armor that the hero can buy and their respective prices

else

hero can sell

Display armor that the hero can buy and their respective prices

hero can’t buy

can\_buy