Henrique Custodio (#101497015)
Comp2152 (50336)- Assignment 2 - Part 2: Explain Your Code (50%)
Due Date: March 23rd 2025

1. **How have we used classes for our project to reuse code?**

We refactored the project to use a base (parent) Character class with common attributes such as combat_strength and health_points, along with shared getter and setter methods for those attributes. We then defined (child) classes, Hero and Monster, which inherit the base classes methods thus allowing us to avoid writing duplicate code and use polymorphism to simplify how we can interact with the created objects during our game. This refactoring makes the code more modular and easier to extend as needed in the future.

2. Provide 1 line of code, as one of many examples, where code is shared between the Monster class and the Hero class?

```
print(f"Hero Strength: {self.combat_strength}, Monster Health: {monster.health_points}")
```

In this line the code the base (parent) Character class attributes are shared between the two derived (child) classes, Hero and Monster, as the line makes use of the combat_strength and health_points attributes.

3. What is the benefit of using complex getters and setters?

The benefit of using getters and setters is how they provide a more controlled access to class attributes. They restrict how attributes can be accessed and set on the object created, which prevents the invalid/unwanted changing of properties - greatly improving code safety. They also allow for simpler error and input checking, which can stop the game from crashing due to invalid values provided by the user affecting object properties. Lastly, it improves debugging by providing easy to read and understand logic for actions interacting with object properties, and by centralizing validation logic for object property values.

4. If we didn't use try-except blocks, what would be the problem?

If we did not use try-except block, there could be issues with the game crashing due to unhandled runtime errors such as a ValueError. For example if a non-integer input was provided for the dream levels, such as a string, this would result in a runtime error and the program terminating due to not being able to handle the exception gracefully. Instead with the use of try-except we are able to more easily prompt the user to try again with a valid input.

5. 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 use the version of python to prevent errors by using the platform module and method calls such as platform.system() and platform.python_version() to check if the environment in which the user launches the game is using a supported python version. For example, we can ensure they are running on Python 3.8 and above, or at the very least using Python 3.6 and above to ensure features such as print f strings and decorators are supported. We

can issue warnings for older versions of Python as well to let the user know that an update would be required or to provide any further reasoning as needed.

6. What's another piece of information we could save inside of the save.txt file?

Another piece of information we can also save inside of the save.txt file would be the total number of games played or the total number of stars earned in previous games. We can also track more detailed information such as the statistics of previous heroes and monsters, such as what the previous hero's health_points and combat_strength was along with that of the monsters they encountered throughout multiple sessions.

7. New Feature:

a. Think of 1 new feature you can add to the game that could use list comprehension and nested if-statements. For now just write 1 sentence that describes the feature:

Now add your new feature description here:

We will create a Party Character Generator that allows the user to create up to 4 characters and roll for a unique "awakening-sign". The awakening-sign grants a specific buff or debuff depending on which sign is rolled and the corresponding stats of the character it is being applied to.

b. Give the new feature you created a short 2-3 -word a title:

Now write your Title here:

Party Character Generator

c. 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

i. Using a list comprehension loop:

The user will be given prompts to create characters at the start of the game by providing them with a name, and after their health and strength is generated they will also roll for their 'awakening-sign'. The user will be allowed to create up to 4 characters in this manner. The user inputs along with character created, their awakening signs, etc; will be kept in lists which will then be read through and saved on the save.txt file via a list comprehension.

We can simply visualize the arrays that will be created and read to construct a proper save.txt file via the following table:

ID	Name	Awakening-Sign	Strength	Health
0	Chris	Dragon	4	11
1	John	Slime	2	5
2	Bob	Eagle	5	12
3	Jill	Wolf	3	16

ii. Using nested conditional statements:

After generating each character, we apply buffs or debuffs based on the awakening-sign. This step will use nested if-statements given the signs will be conditional and give different effects depending upon the characters rolled health_points and combat_strength (Eg. Dragon sign grants +2 Strength if base Strength is ≥3, otherwise only +1).