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?
   We use the Character parent class to create the Hero and Monster class, reusing code by defining the relevant attributes in Character and not needing to repeat that step for the children classes

2. Provide 1 line of code, as one of many examples, where code is shared between the monster class and the hero class?
   monsterObj.health_points and heroObj.health_points

3. What is the benefit of using complex getters and setters?
   Using complex getters and setters, instead of regular getters and setter, allows more checking of inputs. For example, with the health_points setter, we can check if the input is a valid number beforehand

4. If we didn't use try-except blocks, what would be the problem?
   We would not be able to catch any exceptions should it happen, stopping the program. Or we would not know what the exception specifically was about, decreasing transparency and ease of use

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 can check if the python version installed in the player's machine is the latest. If it is not, we can suggest that they update their python version.

6. 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 what type of items the player used and equipped in the previous game

7. **New Feature:**
    a. 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:
    Add spells so that the hero can choose which method of attack to use/heal and allows the monster to have resistances and weaknesses to spell types

---

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

---

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

Now write your Title here:
Spell Centric Fights

Examples:

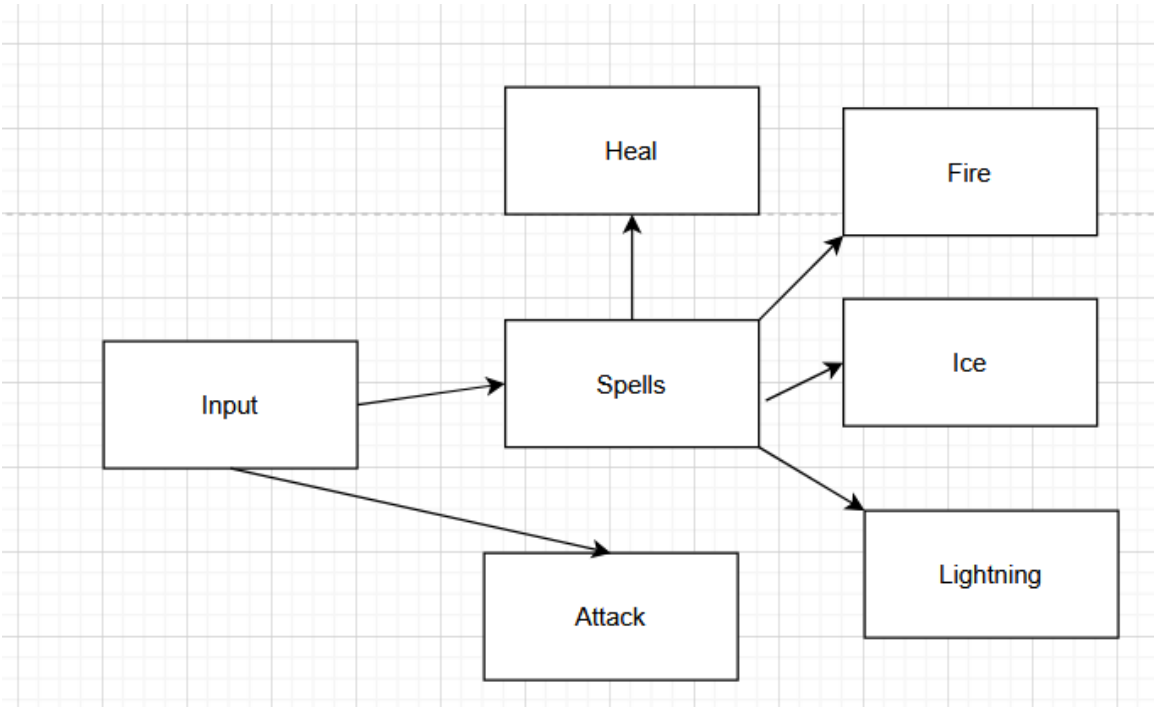
eg a) Multiple Monsters
eg b) Roam Towns
eg c) Dog Scout

---

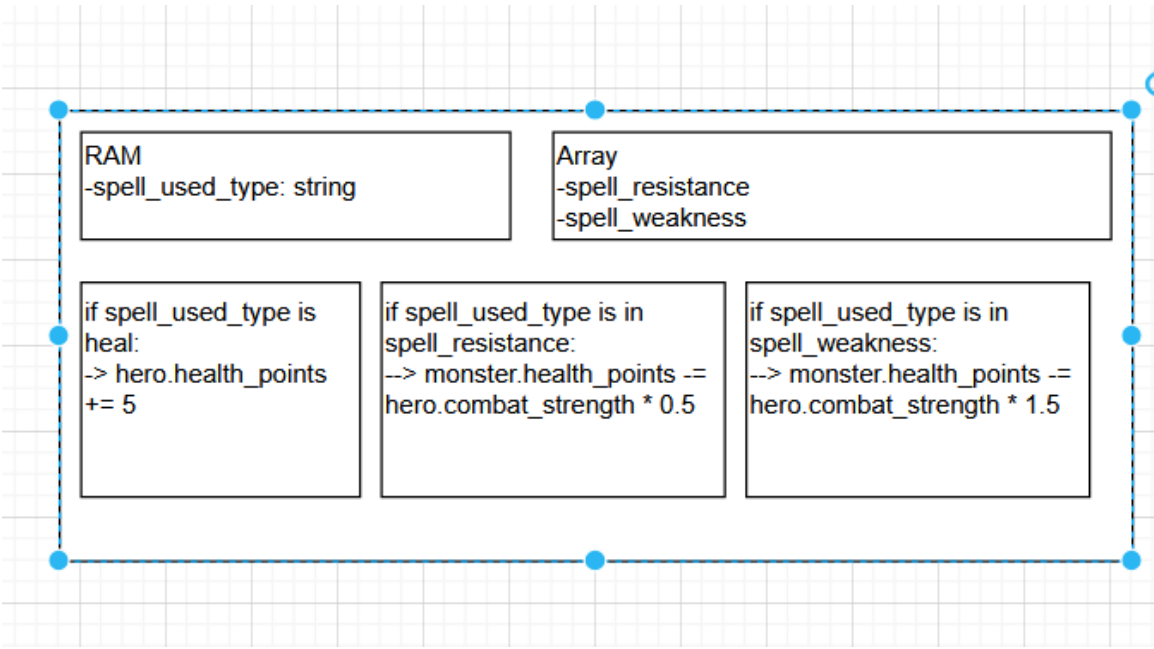
c. Explain how you could implement the idea you chose. You must explain how you would use both of the control structures below. Draw a diagram, map, sketch for each (you can use any software for this, e.g. Draw.io). You don't have to match the style of diagram I have here, just use a visual to describe your idea. Note, you must have loops and conditional statements diagrammed below as needed:

---

i. **Using a list comprehension loop**
Every time in the fight loop, ask the user to input either "Attack" or "Spells". If the user input "Attack", we attack as before. If "Spells", we list a list of spells and ask the user to choose one.
Diagram:



ii. **Using nested conditional statements**
For example, the monster is weak to an Ice Spell and resistant against a Fire Spell. Both the weakness and resistance can be a list. **If** the hero uses a heal spell, the hero HP is restored by 5. **If** the hero uses an Ice spell, the combat strength afflicted is increased by 1.5x. **If** the hero uses a Fire spell, the combat strength afflicted is halved. **Else**, the combat strength afflicted is normal
Diagram:



Example:

eg b) Roam Towns

i. **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)

ii. **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.