

# Parsa molahosseini

101491591

## 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've developed a parent Character class that holds shared properties and methods for our Hero and Monster classes. By using inheritance, we can avoid repeating code for important features like combat strength and health points management. This makes our code easier to maintain and less cluttered, as we only have to define properties and behaviors once in the parent class. It's a smart way to keep things organized and efficient.

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

```
super().__init__(combat_strength, health_points)
```

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

Getters and setters effectively manage access to private class attributes. They enable validation and formatting, protecting data from direct manipulation and improving code maintainability. This approach also allows you to update the class's internal workings without

disrupting dependent code

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

Without try-except blocks, our program might crash if a user enters something unexpected, like a letter instead of a number for dream levels. Using try-except lets us catch those errors and provide helpful feedback, keeping the program running smoothly and improving the user experience.

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.

To ensure smooth screen clearing, we can check the operating system name. If `os.name` is 'nt' (Windows), we use the `cls` command; for Unix, Linux, or macOS, we use `clear`. This prevents any command mix-ups and keeps everything running smoothly.

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 could keep track of the highest combat strength a player has ever achieved across all games. This way, we can monitor how players progress over time and even unlock special abilities or content when they hit certain milestones. It would really enhance the gaming experience by giving everyone a fun sense of long-term growth

## 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:

A dynamic weather system that affects combat strength and health points of both hero and monster based on various environmental conditions, with weather events generated using list comprehension and nested conditionals to determine specific effects.

---

### 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:

**Title:** Dynamic Weather System

---

---

### Examples:

**eg a)** Multiple Monsters

**eg b)** Roam Towns

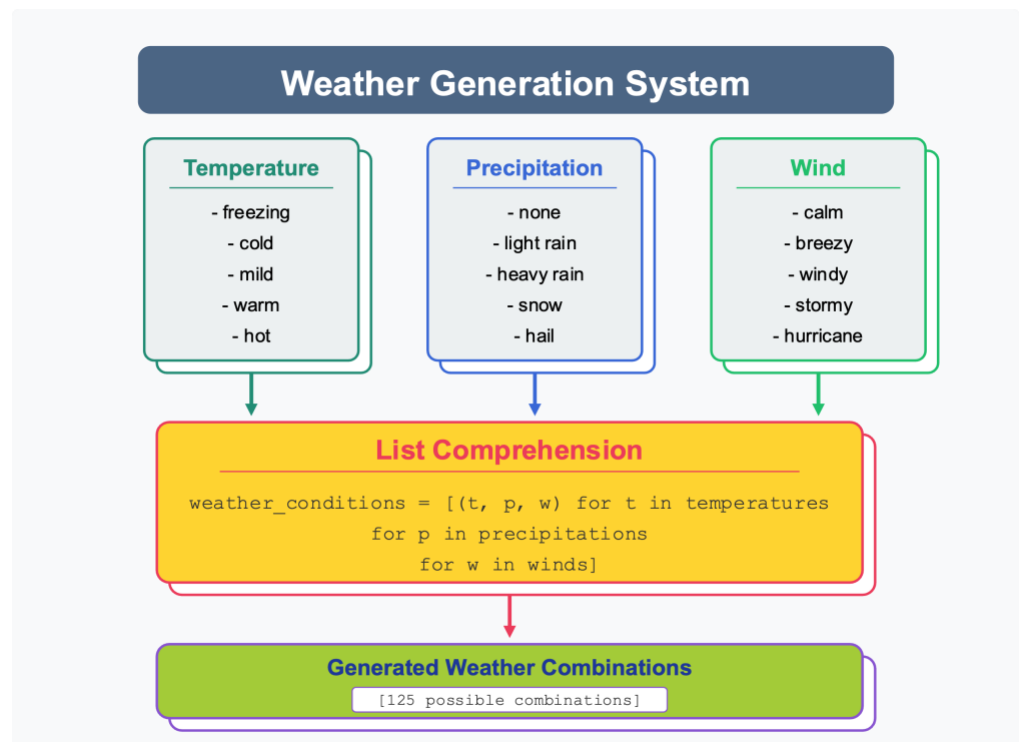
**eg c)** Dog Scout

---

- 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**

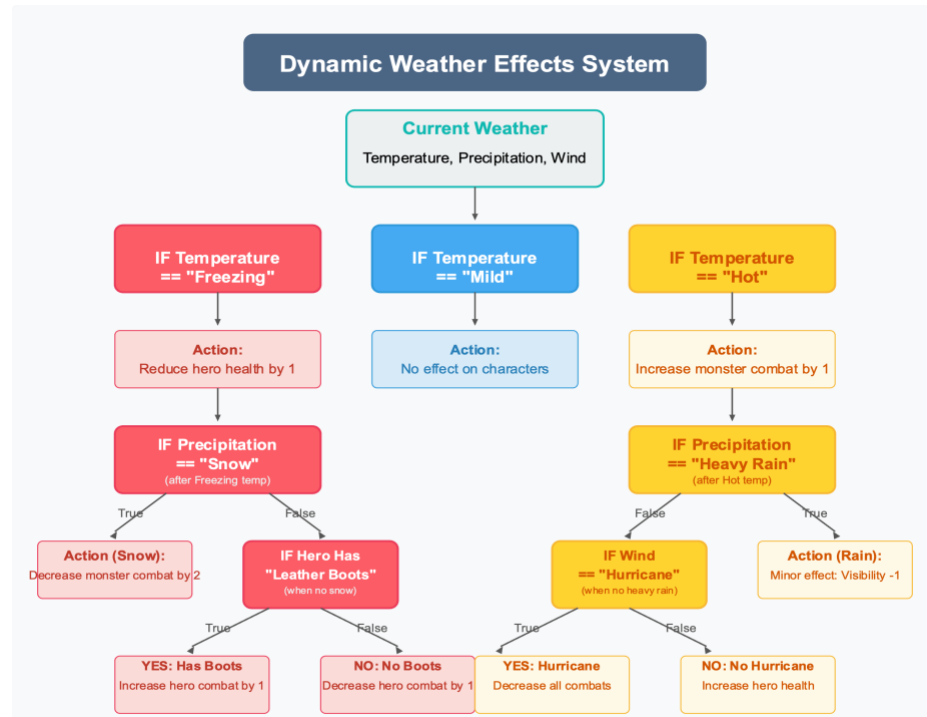
Generate all possible weather conditions by combining different elements (temperature, precipitation, wind) using list comprehension. Each turn, randomly select a weather condition from this list and apply its effects. For example: `weather_conditions = [(t, p, w) for t in temperatures for p in precipitations for w in winds]` would create all combinations of weather. The hero's location could also influence which subset of



ii. **Using nested conditional statements**

Apply different effects to hero and monster based on the current weather condition using nested conditionals. For example, if the temperature is

freezing AND precipitation is snow, THEN reduce monster's combat strength by 2, but if the hero has "Winter Boots", THEN increase hero's combat strength by 1 instead of decreasing it. These complex conditions create strategic gameplay as different equipment becomes more valuable in different weather.



### 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.

