OOP HURDLE TEST – TASK 2

Name: Nguyen Duc Manh

ID: SWH02701 – 105547489

1. Question 1:

- Principle of Polymorphism: allows objects of different classes to be treated as if they were of the same superclass, enabling a single interface to represent multiple types. This makes code more flexible and easier to extend.

- In task 1, both File and Folder instances inherit from Thing. In the Main method, when the same method Print() is called for File and Folder they will give different output. Or when the Main called Size(), for File instance, it will return its name, extension and size in bytes. On the other hand, for Folder instance, it will return the folder's description as well as listing out all the files it contains.

2. Question 2:

- We still need to keep both classes FileSystem and Folder because FileSystem will be the manager to the overall structor, its job is to print out everything we have created as well as organize created instances better. While Folder is used to distinguish to File instance that can hold subfolder(s).

- Keeping them separate follows the Single Responsibility principle and makes the design clearer and more maintainable.

3. Question 3:

The class name Thing is too common, and it may encounter error or misunderstandings when we scale up the program. I suggest to name it "DataEntry" for more clearer and prevent from duplication.

4. Question 4:

- Principle of abstraction: hiding complex details and showing only the essential features of an object. It helps reduce complexity and focus on what an object does, not how it does it.

- I would use it to create an abstract class that can generalize the function between FileSystem and Folder: store other data entries and can print everything that has been added (like a File).

5. Question 5:

Pass and Credit tasks I have submitted to Canvas that implemented both Polymorphism and Abstraction are:

+) 6.1P

+) 7.2C