1: Inheritance - choose your objects

What objects can you think of that are subtypes of other objects?

Animal → Dog/wolf/

Animal → cat/lion/tiger

Charactor → human/male/female/child

a few examples of parent/child relationships

2. Polymorphism - choose your behaviors

think about the types of behaviors those objects could perform in your program, and how those behaviors might “morph” depending on the object subtype.

All animals might have a Jump() function, but a cat might have different jumping capabilities than a dog.

All humans walk/run but might have different capabilities between a male, female or child

As you know, polymorphism also refers to the fact that methods of the same name can be overloaded with different parameter types to have many forms (“poly” + ”morphs”). You might also consider the different parameter inputs one of these methods could take. For example, a Jump() function could take the max height of the jump [Jump(float maxHeight)] or the force applied to it [Jump(Vector3 jumpForce)].

3. Encapsulation - choose your data

Encapsulation is all about protecting data in a class so that it cannot be misused and break your application. Think about which data you want to protect in this way, while still allowing other classes to get or set it. For example:

For an animal, you may need to get or set its name, but prevent the name from being too long.

For a human you can set its name to be restricted and a set length

4. Abstraction - choose your functions

Abstraction will probably be the easiest concept to implement once you are already coding, but it may be helpful to think of some of the high-level functions you want in your project before getting started. For example:

For animals/humans, you might have Walk(), Jump(), Talk() functions.

If you are still feeling lost with no idea what to do for your project, here’s an idea for an extremely simple project:

* There are three different colored shapes that inherit from a base “shape” class (inheritance).
* When you click on each shape, it prints something different to the screen through an overriden DisplayText() function (polymorphism).
* Each shape contains data, such as its name or color, which are stored as properties with getters and setters (encapsulation).
* Code is organized in a way that reduces duplicate code through higher level methods (abstraction).