Create object of the right class for the right problem from a bank of multiple classes/subclasses.

Creational tends to hide what is happening and how behind “factories” and abstract classes.

They tend to focus on the creation of objects and conceal that by using factory classes to create them and hide the details from the class creating them.

Factories and singletons make up most of the common creational design patterns.

A singleton class is a class that has a single object that has all the data and functionality that are likely to be needed. It is referenced by other classes

**Behavioral design pattern**

Focuses on how objects communicate with each other.

Behavioral object patterns use object composition rather than inheritance. Some describe how a group of objects

Composition is another type of relationship between classes.

Instead of the “IS A” relationship of inheritance, it’s “HAS A”

A field will hold a reference to another object using another object to provide some or all its functionality.

In the animal examples – the main class is composed of the animal classes; it doesn’t come from its own utility.

**Structural design pattern**

Rather than use larger self-contained classes, build a program or functionality from smaller classes.

Often new structures are created through multiple inheritance. One class inherits from more than a single parent class to create a new structure.