Object Oriented Programming

Objects:

An entity with a state and a known behavior.

```
Legs = 2; //state
bool isHungry( appetite ) { if (appetite < 50) return true; else return false; } //behavior
```

Classes:

A collection of Objects. A logical entity.

```
Class Dog{
        int Legs = 4;
        bool isHungry = true;
}
```

Inheritance:

When one object acqures all the properties and behaviors of parent object:

- provides code reusability
- used to achieve runtime polymorphism

```
Class Character{
        int healthPoints = 100;
        float damageMod = .7;
        float[] coords = {0.0, 0.0};
        bool isDead(healthPoints) { ... }
}
```

```
Class Enemy : Character {
        // inherits all attrs from Character
}
Class Player : Character {
        // inherits all attrs from Character
}
```

Abstraction:

Relates to both Encapsulation and Data Hiding:
Programmer hides all but relevant data for an object to reduce complexity & increase efficiency

Encapsulation:

Encapsulation binds together the data and functions which manipulate that data, keeping them safe from outside interference and misuse. This also led to the concept of Abstraction (Data Hiding); making data visible only to classes with proper permissions.

Polymorphism:

Polymorphism is the ability to present the same interface for different data types:

Example: Integers and Floats are polymorphic in that you may apply many of the same operations to both

The 4 Main 'Principles' of OOP are:

Inheritance, Encapsulation, Abstraction, and Polymorphism