

## **What are the four pillars of Object-Oriented Programming? Explain each pillar.**

The four pillars of OOP are Abstraction, Encapsulation, Inheritance, Polymorphism, also commonly coined "A-PIE". Abstraction is commonly used when making things like objects, classes and variables and this provides a sense of simplicity when there's much more underlying code that is more complex. Encapsulation is keeping specific fields private and inaccessible but providing access to them with public methods, thus making data safer to be handled. Inheritance allows you to create new classes that share attributes of other existing classes and finally Polymorphism allows for overloading or overriding methods. Essentially allowing two pieces of code to mean two different things or do two different things.

## **What are the differences between abstract classes and interfaces? When should you use one over the other?**

The primary difference between abstract classes and interfaces is that abstract classes can also have concrete methods. What this means is, when you're implementing an interface, it requires you to also implement all the methods used in said interface, while abstract classes can have their own methods and keep those separate from the implementing class. An abstract class can also have public, private or protected variables, while an interface is by default, public. I would say the most common time you want to use one over the other is when you're trying to control more of what is accessible to other classes, abstract classes will allow you to have private variables that are not accessible to the rest of the classes.