

Week 05 Research

Object-Oriented Program: The Four Pillars

Questions:

What is the relationship between a Class and an Object?

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

What is the relationship between a Class and an Object?

So, what is the relationship between a Class and an Object? The website Geeks for Geeks states that the class is a user-defined data type. That's technically true; however, I have always considered the class as a set of blueprints. A class defines and classifies the properties and methods needed to manipulate the object. In many ways, a class is a contract with the object that declares the required data and methods to manipulate its state and behavior. Classes are a representation of a tangible real-world object that we create in code.

Object! Yes, an object is the concrete implementation of the class. Once the class is instantiated, it builds the brick-and-mortar entity from the blueprints, allowing us to manipulate code. To quote the Geeks for Geeks article, "An object has an identity, state, and behavior," It can take on its own life in code.

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

- Encapsulation
- Abstraction
- Inheritance
- Polymorphism

Encapsulation:

Encapsulation is defined as enclosing or hiding properties or methods within the class object. Therefore, the object can only access the data and methods; no outside code or other objects can call the private methods or access the private data properties. In JavaScript/ECMAScript, encapsulation is not strictly implemented. JavaScript best coding practices recommend using a backing field for private data properties (i.e., this._name) and then exposing the property through getters and setters.

<https://www.freecodecamp.org/news/four-pillars-of-object-oriented-programming/>

[https://www.techtarget.com/searchapparchitecture/definition/object-oriented-programming-OOP#:~:text=Object%2Doriented%20programming%20\(OOP\)%20is%20a%20computer%20programming%20model,has%20unique%20attributes%20and%20behavior.](https://www.techtarget.com/searchapparchitecture/definition/object-oriented-programming-OOP#:~:text=Object%2Doriented%20programming%20(OOP)%20is%20a%20computer%20programming%20model,has%20unique%20attributes%20and%20behavior.)

<https://www.geeksforgeeks.org/introduction-of-object-oriented-programming/>

<https://www.freecodecamp.org/news/object-oriented-programming-concepts-21bb035f7260/#:~:text=The%20four%20principles%20of%20object,scary%20for%20a%20junior%20developer.>

<https://www.digitalocean.com/community/conceptual-articles/s-o-l-i-d-the-first-five-principles-of-object-oriented-design>

Week 05 Research

Abstraction:

Abstraction relates to encapsulation in that abstraction only expose the data properties and methods needed to use the object. I know this is the technical definition; however, I have conceptualized abstraction with abstract or base classes. The abstract class provides data properties, methods, and virtual methods to be inherited by the derived class. So, for example, in a virtual method that implements printing, each derived class would override that method with how it handles printing.

Inheritance:

In the article, How to explain object-oriented programming concepts to a 6-year-old, Alexander Petkov boldly states, "Objects are often similar. They share common logic," this is the perfect scenario for inheritance where one class inherits properties and methods from the other. This concept extends the functionality and makes the code reusable. When should you use inheritance? I follow the Liskov Substitution Principle, which states that if you could replace your child with the parent class, then it is a good use of inheritance.

Polymorphism:

Polymorphism uses an override on method so that the child class implements its requirements. The classic example is a parent class shape with a method calculating area. Circle, Square, and Triangle are all child classes with different calculations for the area method. The child classes would override the area class and use the appropriate calculation.

<https://www.freecodecamp.org/news/four-pillars-of-object-oriented-programming/>
[https://www.techtarget.com/searchapparchitecture/definition/object-oriented-programming-OOP#:~:text=Object%2Doriented%20programming%20\(OOP\)%20is%20a%20computer%20programming%20model,has%20unique%20attributes%20and%20behavior.](https://www.techtarget.com/searchapparchitecture/definition/object-oriented-programming-OOP#:~:text=Object%2Doriented%20programming%20(OOP)%20is%20a%20computer%20programming%20model,has%20unique%20attributes%20and%20behavior.)
<https://www.geeksforgeeks.org/introduction-of-object-oriented-programming/>
<https://www.freecodecamp.org/news/object-oriented-programming-concepts-21bb035f7260/#:~:text=The%20four%20principles%20of%20object,scary%20for%20a%20junior%20developer.>
<https://www.digitalocean.com/community/conceptual-articles/s-o-l-i-d-the-first-five-principles-of-object-oriented-design>