Week 5 Research Assignment

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

The four pillars of OOP are:

- Abstraction:
    - Abstraction is coding in a way that you remove the need for overly complex functionality and write the program in a way that only shows the essentials to a user. Abstraction allows developers to create more modular and maintainable code, as they can separate the implementation details from the interface. It also makes it easier for other developers to understand and use the code, as they do not need to know all the implementation details to use the program effectively.
    - **Example:** If you have a class for a car, you might have methods for starting the engine, accelerating, and braking. The user of the car object does not need to know the details of how the engine works or how the brakes are applied. They just need to know how to use the methods to operate the car.
- Encapsulation:
    - Encapsulation is encapsulating similar or like code together and limits how much code is accessible or visible. You can use classes to create or store private or public properties and methods.
    - **Example:** In a class for a bank account, the account balance might be a private property that cannot be accessed directly from outside the class. Instead, the class might provide public methods for depositing and withdrawing money from the account.
- Inheritance:
    - Inheritance lets an object access or acquire the methods and properties of another object. This allows for better reusability as it is common to have code that functions similarly with a small change or two. Inheritance allows there to be parent and child classes that share similar information or functionality.
    - **Example:** A class for a vehicle, which has properties such as "make", "model", and "year", as well as methods such as "start" and "stop". A subclass for a car can inherit all of these properties and methods from the vehicle class, but can also add its own properties such as "number of doors" or methods such as "play music".
- Polymorphism:
    - Polymorphism is another concept in object-oriented programming that refers to the ability of objects of different classes to be treated as if they were objects of the same class. This allows for greater flexibility and reusability of code, as the same method or function can be used with different types of objects.
    - **Example:** You could have a parent class of “Shape” with child classes of “Rectangle”, “Square”, and “Circle”. Each would implement a method of “Area”.

Each child object would be able to use the method of “Area” without needing to adjust the code for each shape.

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

2. What is the relationship between a Class and an Object?

- Classes are blueprints for creating objects in JavaScript. Classes allow for developers to be able to build several objects that will all follow the same basic structure but allows them to have their own values for their properties. Classes allow developers to be able to create multiple objects that have the same properties and methods without having to write new code every time. Allow for more modularity, reusability, and ability to maintain the code.
- URL: https://www.geeksforgeeks.org/classes-and-objects-in-javascript/#