Week 5 Research

Rachel VanHorn

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

The four pillars of OOP are abstraction, encapsulation, inheritance and polymorphism. Abstraction means to hide the complexity of the code you are writing, you can do this by keeping your code concise and succinctly written. Encapsulation means hiding the details of exactly how something works. In the case of OOP, that means that your objects should control their own state. If you need to make changes to your object at some point in the future, you can just make one change in the objects code and it will apply everywhere the object is used. Inheritance means that one object can inherit properties from another (parent) object. It lets us use code that is already written so we don't have to keep repeating code that is similar across objects. Polymorphism means that objects that share an inheritance chain do not have to be completely identical. They can still have individual aspects based on the specific object. You can have multiple children that inherit something from a parent object, but they can all have individual aspects that are unique to them.

Sources:

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

https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/Object-oriented_programming

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

Classes are templates or blueprint used to create Objects. The object that is created from the blueprint a class lays out is an instance of the class but is a unique object that can utilize the methods inside the Class. You can use one class to create many different objects. You can also have a parent class that has multiple child classes that derive all or some of their code from the parent.

Sources:

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Classes

https://www.geeksforgeeks.org/classes-and-objects-in-javascript/

3. What is an exception and what are best practices for handling them?

An exception is an error that happens when code is running. Errors can be because of a mistake made by the programmer and they can also be from unexpected inputs from users or issues with something you are waiting on from another source. The best practices for handling exceptions are to use "Try, Catch, Finally" statements. You can use these when you have code that might have unexpected errors come back from user input or other websites not functioning properly. You put the code you want to try in one

block, you put a catch statement in the next block that will catch errors and then you have a finally block that will run no matter the result of the “Try” and “Catch” blocks.

Sources:

https://www.w3schools.com/js/js_errors.asp

https://www.tutorialspoint.com/javascript/javascript_error_handling.htm