Carrie Scott

7/31/18

**Object Oriented Programming**

**Problem**

With procedural and functional programing, functions are dependant on the values stored in them. As programs grow, lines of code are often copied and pasted throughout, creating room for error. The interdependence between functions and their values means that one broken function can have a rippling effect.

**How this is solved with Object Oriented Programming**

With OOP we combine related functions and variables into objects. This is called **Encapsulations** because we are packaging the function and variable together. This allows us to write functions without parameters, which makes them easier to reuse and maintain. We call the functions “methods” and the variables properties.

**Abstraction** is a feature of OOP meaning that the complexity of our functions and variables are tucked away, making them easier to understand and limiting the impact of any one object changing.

With **Inheritance** we eliminate redundant code by allowing elements to inherit properties.

**Polymorphism** is a feature that allows us to get rid of long switch/case or if/else statements by nesting the rendering method into the object itself.