

## Object Oriented Programming Quicksheet

### **Concept overview**

Object Oriented programming is based on the concept of creating classes and objects. There are benefits to object oriented programming as opposed to functional programming. The ability to create a class and have specialized methods that pertain to that class is incredibly powerful, it makes the flow of the program easier depending on what you are doing in your program. One example I can think of is games, since games use many different assets and objects that appear, change and disappear; it makes sense to program using OOP. If you had to use simple functions in a game it would be quite challenging, you would need to write and constantly call functions. This would be inefficient because you would need away of “attaching” the function or calling it for each object. This would also lead to large amounts of functions which would not really be manageable in the long term. OOP also lends itself to modularity, If I want to apply my class method to a new object it's extremely easy to do that.

### **How OOP could fit into my teaching context**

I want to explore utilizing OOP in my regular computer science class, I am not experienced enough to implement this yet however. I do use this in my game development class when using unity 3-D, it makes much more sense to me in this context. I know that OOP is critical for CS students to understand and I can identify sometimes when it might be beneficial to use OOP over functions. One day I may plan to teach AP CSA which would make it necessary to understand OOP in a deeper way.

### **How I might teach OOP**

I will continue to explore OOP through game development, I will need to think of specific projects to use outside of the context of game development. My students use github, BASH, Python, JavaScript and Arduino, we have many options to implement OOP.