This word document covers the explanations of vital OOP principles - Encapsulation, Inheritance, Polymorphism, Abstraction. These are the most important concepts to object oriented programming. The concepts will be covered in the order they were introduced.

Encapsulation is the act of protecting code by the act of making objects (classes) private/protected - this requires the explicit usage of Get/Set operators, meaning the programmer has less room for mistakes as they must be intentionally deliberate with how they use classes. This also prevents other classes from being able to easily access the attributes of other classes. A popular real world example of encapsulation is a car; all the components work together, but only the parts that need to – the steering for example is completely independent of the other mechanisms of the car and do not affect it.

Inheritance, is the concept of an object inheriting the attributes/methods of another class, forming a parent-child relationship of sorts – it will share the same variables/methods of its parent class, but can still have its own unique attributes. This promotes code reuse which is a prime feature of the OOP paradigm. Inheritance creates a class hierarchy where the parent classes are usually more abstracted to cover all possible attributes the object possess, while the children classes will usually cover the specifics of each exact object.

Polymorphism is the concept of a method/class being able to adapt processing other objects depending on their type where multiple implementations of one method are written are usually known as overloading a method – where you call the same method by name, but the method behaves differently depending on the object passed in its parameters. Polymorphism also takes the form of abstract classes that are very malleable – take the example of creating a class to cover shapes – this shape class can be used to create a wide variety of different objects such as squares, circles, etc. hence being polymorphic in nature.

The final concept that is vital to OOP is abstraction – Abstraction is the concept of hiding information that is ultimately useless to the end user and only showing the relevant details. Abstracting classes ultimately makes them easier to understand and implement. Abstracting is related to encapsulation as it is a form of abstraction. Abstraction is prevalent in all fields of programming including websites – for example, when you login your account onto a website, you only have to enter the relevant details and then you are took to the main page signed in – the process of verifying your account details against the database is hidden from the user as you don’t need to know this as an end user.