**Assignment01-Q3:** The benefits of using Maven include building projects in various different environments without the need much configuring or hassle, the automation of dependency downloads, and streamlined project management, making testing, development, and release much more efficient.

**Assignment01-Q4:** The *Open-Closed Principle* states that software entities like classes, modules, and functions, should be open for extension and closed for modification. This means that existing code should not be altered, but new behaviour can be added by extending existing code or employing interfaces and abstract classes. This principle ensures the flexibility and maintainability of code and minimizes the impact of new requirements on the existing code, such as the risk of new bugs.

The *Information Hiding Principle* states that objects should hide certain information from other objects. This is an important principle as it means that some information that should be private, like personal information or passwords, is inaccessible by other objects and only available through certain means, like a method that verifies its being accessed by an appropriate object before returning the private data. This principle overall reduces system complexity and increases security and maintainability.

The *Self-Documenting Principle* says that code should be written clearly and so that it is easily understandable on its own, not requiring excess comments or utilizing overly-complicated structures, names, etc.. This principle makes it simpler for developers to read and understand the code, thus making modifying and testing code easier, and so the overall development process is more efficient.

The *Single Responsibility Principle*, that each class should encapsulate just one functionality, helps make maintenance easier and improves readability, since changing one aspect of the system will not affect unrelated parts since each functionality is encapsulated in its own class. This reduces the risk of system bugs and makes the development process easier, as developers do not have to search for all the classes that together encapsulate a functionality.