

Assignment01-Q3:

The main benefit of Maven is providing project management features to Java. It allows for easy dependency management, so all required files are available to use within local repositories. It works across platforms and increases performance when building/compiling and allows developers to debug and test easily. The pom.xml file which can also be preconfigured, allowing for easier project creation, while also allowing manual configurability for specific cases, such as using custom plugins.

Assignment 01-Q4:

Divide and Conquer Principle: The divide and conquer principle is useful because the purpose of it is to take a big problem and divide it into smaller problems that are easier to solve. This makes each problem more manageable and improves readability and editability later. All these sub-problems are then combined to solve the main problem.

Encapsulation Principle: The encapsulation principle helps prevent unintended/unnecessary changing of objects by keeping them encapsulated within other objects with other related parts of data. For example, having a private variable within a public class to prevent the variable from being edited outside of the class, and only editing it using methods within the class. This can maintain data integrity and flexibility as it makes future implementations easier. It also helps with debugging as you can edit the inside of the class without affecting code outside it and it makes it easier to read as all relevant data is grouped together.

Interface Principle: The interface principle helps generalize certain methods that are used across various classes (i.e. a “contract”) so that these classes can implement them without overlap. This prevents redundant repetition of code and allows for more flexibility between classes as the methods are abstracted (only necessary information is implemented), allowing for easier changeability and testability between as the “shared” methods as there is less code to be modified.

Single Responsibility Principle: As the name suggests, the single responsibility principle means each class has its own task to accomplish. This goes back to the divide and conquer principle as the problem should be broken down into sub problems, rather than having one class tackle multiple. This simplifies the code and allows for easier editing as there is less coupling and allows for individual testing of each class’s implementation.