Explanation:

I have made this system for doing operations on various types of Matrices with the help of a parent class Matrix and from it inherit several subclasses like Square Matrix, Rectangular Matrix, Row Matrix etc..

By making these classes my code has been able to generalise the functions being called from them. In some Matrix classes for eg: Diagonal Matrix in which which we can optimise the formation of their objects and storing them, we have taken an array into it rather than taking a Matrix into it like in the case of Square Matrix.

This example is one of many instances where the code has been optimised and hence to make the functions in them exhibit the functionality of functions stated in the (parent) Matrix class they have been Overrided accordingly.

In this way we are able to generalise our coding as well as optimise it wherever needed.

In the functions where we know optimisation is not so much possible there we have made the use of super keyword to implement the functioning of the functions stated in the (parent)Matrix class.

The principle of OOPS which I used to help me with programming this code efficiently was majorly inheritance as well as polymorphism, as all the _Matrix classes like Scalar_Matrix class or SquareMatrix class could inherit from the (parent)Matrix class. And since they do so the methods can be overrided according to their own classes description. Since all the _Matrix classes extend from the (parent)Matrix class we can see the usage of polymorphism when the objects are created of different-different Matrix types taking the (parent) Matrix class as reference.