Exercise 1

i) • Divide and Conquer - Break a problem into smaller, manageable objects. Each object handles a specific part of task.

• Encapsulation and modularity - combine data and behavior into single unit (object) and ensure each object is self-contained with a clear purpose

• Public interface - Define how other objects can interact with an object by specifying a clear and limited set of accessible methods

• information hiding - Hide the internal workings of an object and expose only what's necessary, shielding users from complexity.

• Generality - to design objects to solve a type of ~~mask~~ task rather than specific instance, enchancing reusability.

• Extensibility - allow existing object or system to be extended with new behaviours without modifying the existing code.

• Abstraction - focus on essential features of an object and ignore irrelevant details, simplifying probkm solving.

2) i) code reusability - object and classes can be reused in different programs, reducing development.

ii) improved maintainability - encapsulation and modularity make updates and maintenance easier.

iii) ~~cost and time efficiency - objects and classes can be reused in different programs~~

iii) cost and time efficiency - saves resources by leveraging existing tested components.