



4.4.1 Abstraction & Automation

Definitions

Algorithm

- A sequence of steps that completes a task in a finite amount of time.

Representational Abstraction

- A representation arrived at by removing unnecessary details.
- Example: A map of the roads without showing the buildings and trees.

Abstraction By Generalisation

- A grouping by common characteristics to arrive at a hierarchical relationship of the 'is kind of' type.
- Example: A monkey is a kind of mammal.

Information Hiding

- The process of hiding all details of an object that do not contribute to its essential characteristics.
- Example: Private attributes are hidden.

Procedural Abstraction

- Represents a computational method by abstracting away the actual values used in a particular computation.
- The result is a **procedure** that can be used with different values..

Functional Abstraction

- The process of simplifying a problem by hiding the computation method.
- Only knowledge about the function name, its parameters and return value need to be known to use it.

Data Abstraction

- Data abstraction is a methodology that enables us to isolate how a compound data object is used from the details of how it is implemented.
- Example: We know a stack is a LIFO data structure that has operations to push and pop. The implementation could be achieved in different ways. It could be implemented as an array and a pointer for top of stack.





Problem Abstraction

- Details are removed until the problem is represented in a way that is possible to solve, because the problem reduces to one that has already been solved.

Decomposition

- Breaking a problem into smaller sub-problems, each of which solves an identifiable task, each of which might be further subdivided;

Composition

- The process of combining simpler components together to form a more complex solution.
- Example: Combining procedures into compound procedures.
- Example: Combining data objects to form compound data.

Automation

- Models are put into action to solve problems.
- This is achieved by:
 - Creating algorithms
 - Implementing the algorithms in program code (instructions)
 - Implementing the models in data structures
 - Executing the code.

