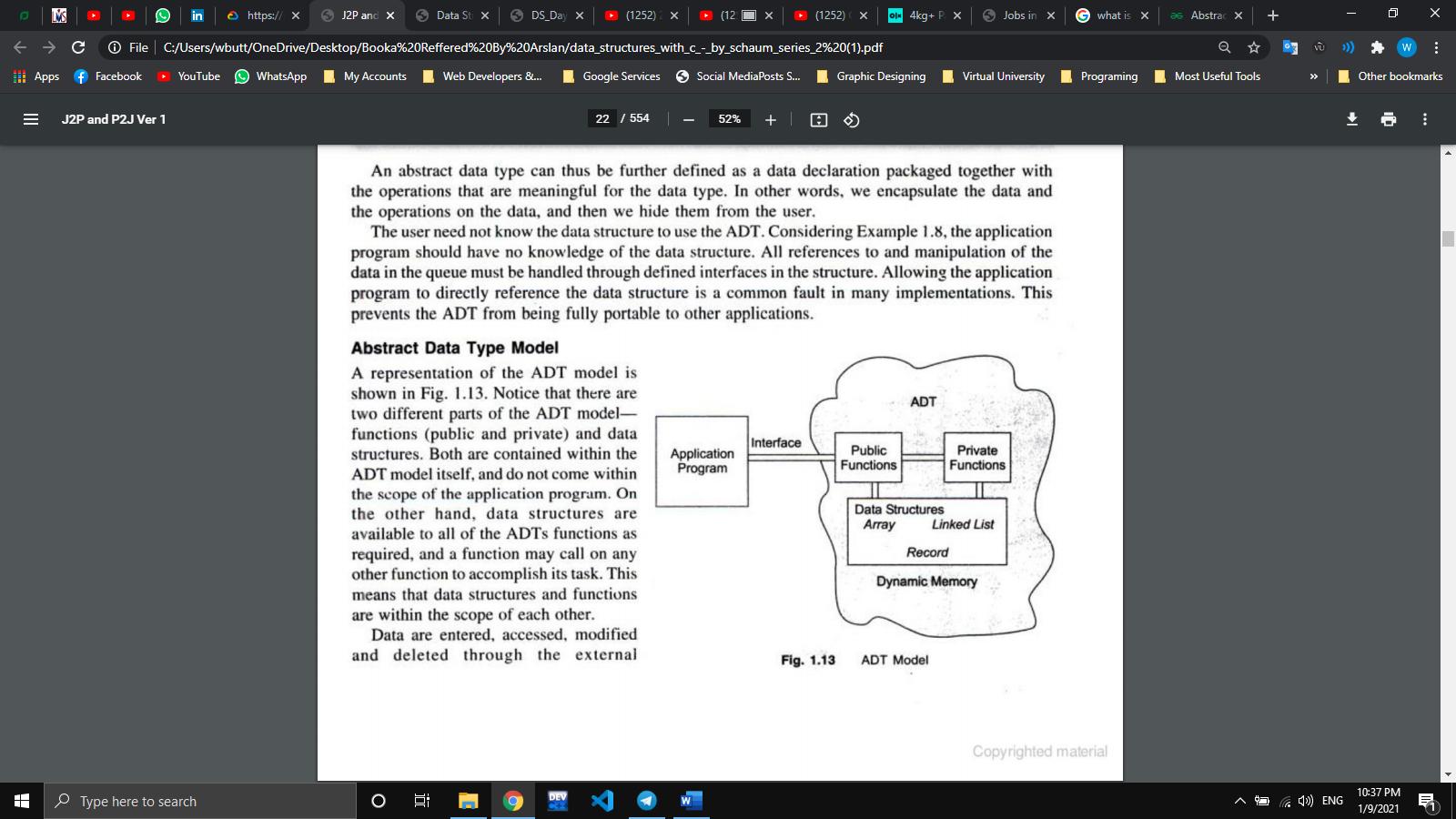
**DATA STRUCTURES**

**Day 2:**

**Data Abstract Data type (ADT):**

Abstract Data type (ADT) is a type (or class) for objects whose behavior is defined by a set of value and a set of operations.

The definition of ADT only mentions what operations are to be performed but not how these operations will be implemented. It is same abstract classes. And we use these techniques Abstraction and encapsulations.

In the above program, get’s the functions but its implementation is hidden (user don’t know about implementation, which data structure are used inside the program etc.)

**Algorithm: Complexity, Time-Space Tradeoff**

An algorithm is a well-defined list of steps for solving a particular problem. Algorithm is more efficient, using less time and less memory(space) usage.

**Complexity** is a function which tells required time & memory for a given algorithm.

Note: Every algorithm has a particular data structure. These are some Important Algorithms:

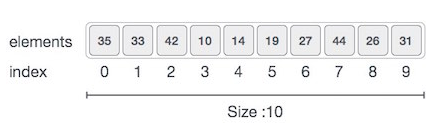
* ***Searching Algorithm***
* ***Linear Search***
* ***Binary Search***

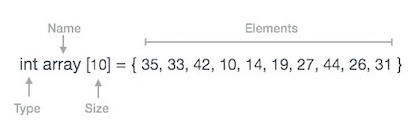
**Arrays:**

An array is a collection of items stored at contiguous memory locations. The idea is to store multiple items of the same type together. Array is a linear data structure and available in all programing languages.

***Element*** − Each item stored in an array is called an element.

***Index*** − Each location of an element in an array has a numerical index, which is used to identify the element

**Representation of an Array:**



**Basic Operations**

Following are the basic operations supported by an array.

* ***Traverse*** − print all the array elements one by one.
* ***Insertion*** − Adds an element at the given index.
* ***Deletion*** − Deletes an element at the given index.
* ***Search*** − Searches an element using the given index or by the value.
* ***Update*** − Updates an element at the given index.

**Dynamic Array:**

Dynamic arrays are used when we don’t know about the size of our array, and we have to get the size from the user at compile time.

**! Task Completed**