- **Class: A blueprint for creating objects that define attributes and behaviors.**

  Example: In a game, you can have a "Player" class that defines attributes like "name" and behaviors like "move" and "attack."

- **Object: An instance of a class that encapsulates data and methods.**

  Example: An object of the "Player" class could be "player1" with a name "John" and the ability to call methods like "move()" and "attack()".

- **Interface: A contract specifying a set of methods that a class must implement. Focus on can-do what.**

  Example: An "IResizable" interface might require classes to implement methods like "resizeWidth()" and "resizeHeight()".

- **Inheritance: The mechanism by which a new class can inherit properties and behaviors from an existing class.**

  Example: A "Car" class can inherit attributes such as “color”, “door” and methods from a more general "Vehicle" class, like "speed" and "startEngine()".

- **Polymorphism: The ability of objects of different classes to respond to the same method call.**

  Example: Both "Circle" and "Rectangle" classes can respond to a "calculateArea()" method, even though they calculate areas differently.

- **Encapsulation: The bundling of data and methods into a single unit (a class) to control access and protect data integrity. Hiding things that would like not to show or unnecessary to show.**

  Example: Encapsulating a "BankAccount" class allows you to control access to account balance and ensure it's not modified directly.

- **Data Structure: A way to organize and store data efficiently, such as arrays, lists, or trees.**

  Example: An "Array" data structure stores elements in a linear sequence, while a "Tree" data structure organizes data hierarchically.

- **Algorithm: A step-by-step procedure for solving a specific problem or performing a task.**

  Example: A sorting algorithm like "Bubble Sort" rearranges a list of numbers in ascending order by comparing and swapping adjacent elements.

- **Abstraction: Simplifying complex systems by focusing on essential details and ignoring irrelevant ones.**

  Example: Using a TV remote control abstracts the complex inner workings of the TV, allowing users to change channels without knowing the details of how it works.

- **Recursion: A technique where a function calls itself to solve a problem in smaller, similar steps.**

  Example: A recursive function to calculate the factorial of a number, like "factorial(5)", breaks it down into "5 * factorial(4)", and so on until it reaches "factorial(1)" as the base case.

- **Abstract Data Type (ADT): A high-level description of data and the operations that can be performed on it.**

  Example: The ADT "Stack" describes data organized in a last-in, first-out (LIFO) manner with operations like "push" and "pop" to manipulate the data.

Big O Notation: A way to describe the upper bound or worst-case performance of an algorithm in terms of its input size. It's the way we measure the efficiency of the algorithm.

Example: In Big O notation, an algorithm with a time complexity of O(n) means that its runtime grows linearly with the size of the input.