What is Abstraction?

Abstraction in programming is the process of simplifying complex systems by hiding unnecessary details and exposing only the essential features to the user. This allows developers to work effectively without needing to understand the intricate workings of every component, for instance when using the print() function in python, it takes one line of code to print what you want displayed on the computer screen, but it actually takes over 3000 lines of code to get the print function to execute and display what you want on the computer screen. One major benefit of abstraction is it improves code maintainability, by reducing complexity and hiding implementation details, developers can modify or replace components without affecting other parts of their program which allows for easier updates and enhancements.  
In this block of C# code:

{  public void Display()

    {

        Console.WriteLine($"Name: {\_name}");

        Console.WriteLine("Jobs:");

        foreach (Job job in \_jobs)

        {

            job.DisplayJobDetails();

        }

    }

In this example, the Display() method in the Resume class uses abstraction by calling the Display() method of the Job class. This hides the details of how job information is formatted and printed, allowing the Resume class to focus on its purpose without getting bogged down in the specifics of job display logic. Abstraction is widely applied in software development, particularly in user interfaces and APIs. For example, when using a graphical user interface (GUI), users interact with buttons and menus without needing to know the underlying code that executes their commands.