Explaining Abstraction

Abstraction can simply be defined as the way we turn complex ideas into simple codes. It is the first principle in programming with classes. It makes it easy to talk about codes with both programmer and non programmers. In other words, it makes our codes easier to understand. This can be considered as an advantage of abstraction. Abstraction also reduces the complexity of codes and adds up higher level of security. A real-world example of abstraction is the use of methods in object-oriented programming.

using System;

abstract class VideoPlayer

{

public abstract void PlayVideo(); // Abstract method (no implementation)

}

class YouTubePlayer : VideoPlayer

{

public override void PlayVideo()

{

Console.WriteLine("Playing YouTube video...");

}

}

class Program

{

static void Main()

{

VideoPlayer player = new YouTubePlayer();

player.PlayVideo(); // Calls the abstracted PlayVideo method

}

}

Abstraction also involves some other programming terms such as class, instance, instantiate and also method. All this makes it easier to understand programming through clarity in communication