What is abstraction and why is it important?

Abstraction is basically about focusing on what an object or system does rather than how it does it. It hides the unnecessary details so that a bigger project is easier to understand and work with. A good example of abstraction is when you use an ATM. You just push buttons to make a transaction, and you do not need to know what is happening behind the scenes when you press that button. Abstraction works the same way in programming. It breaks things down so that you can do simple tasks that trigger more complex operations running elsewhere in the program.

The benefit of abstraction is that it makes it easier to add new ideas or features without needing to understand every single part of the system. For example, when you press “checking” or “savings” on the ATM, each button connects to a different process, but you only need to know what your button does, not how the entire system works.

Abstraction is important because it enhances collaboration and encourages reuse. The components you create can be used again in other parts of a program or even in new projects. One of its greatest benefits is that it simplifies complexity. Programs can become very large, and tools like abstraction make them more manageable. This makes it easier to find and fix problems when something goes wrong, which saves time, money, and reduces downtime, something that is extremely important in the technology world.

Here is an example of abstraction from my code: public string ToStorageString() and public string FromStorageString()

This is an example because it just calls FromStorageString() to get the object.